LABORATORY FOR COMPUTER SCIENCE

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

MIT/LCS/TR-406

CALVIN: A RULE BASED EXPERT SYSTEM FOR IMPROVING ARRHYTHMIA DETECTOR PERFORMANCE DURING NOISY ECGS

Warren K. Muldrow

September 1987

545 TECHNOLOGY SQUARE, CAMBRIDGE, MASSACHUSETTS 02139

CALVIN: A RULE BASED EXPERT SYSTEM FOR IMPROVING ARRHYTHMIA DETECTOR PERFORMANCE DURING NOISY ECGs

by

Warren K. Muldrow

Submitted to the

DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

April 1987

Signature of Author ......Warren K. Muldrow (RGM)......
Department of Electrical Engineering and Computer Science,
April 30, 1987

Certified by ......Roger G. Mark, MD......
Professor Roger G. Mark, Thesis Advisor

Accepted by ....
Arthur C. Smith, Chairman
Committee on Graduate Students

CALVIN: A RULE BASED EXPERT SYSTEM FOR IMPROVING ARRHYTHMIA DETECTOR PERFORMANCE DURING NOISY ECGS

by

Warren K. Muldrow

Submitted to the Department of Electrical Engineering and Computer Science on February 6, 1987 in partial fulfillment of the requirements for the Degree of Master of Science

ABSTRACT

Human experts far outperform automated arrhythmia detectors in analyzing ECG data corrupted by noise and artifact. Humans make use of considerable *a priori* knowledge about cardiac electrophysiology and knowledge acquired from the specific ECG under analysis. R-R intervals, coupling intervals of ectopic beats, and commonly occurring beat patterns observed during noise-free ECG segments form a knowledge base which is used in accurately detecting and classifying true QRS complexes in the presence of severe noise.

In the present study, we developed and tested an expert system that improves the performance of an arrhythmia detector during noisy ECG data. The system (CALVIN - CALipers in Very Intense Noise), which was developed utilizing a modified version of the YAPS Production System, functions as a postprocessor to an existing state-of-the-art arrhythmia detector (ARISTOTLE). During noise-free segments of data, CALVIN operates under the assumption that ARISTOTLE's beat annotations are accurate and constructs a knowledge base that characterizes the relative beat timing, the beat morphology, and the underlying rhythm of the ECG under analysis. When the noise level exceeds a predetermined threshold, ARISTOTLE's beat annotations become unreliable, while the integrity of the QRS sensitivity remains intact. CALVIN then extracts only the time-of-occurrence, the morphology measure, and the local noise level estimate of each detected event from the data stream. The knowledge base is applied to the extracted information to distinguish and classify true QRS complexes from false positive beat detections.

The expert system was evaluated using 8 tapes selected from the AHA Database to which 2 minute bursts of noise (from an existing noise database) were added. The noisy data was presented to ARISTOTLE operating alone and to ARISTOTLE assisted by CALVIN. The results demonstrate that CALVIN is able to significantly improve the accuracy of beat annotations during intense noise.

Acknowledgements

I would like to thank the following individuals who made the completion of this project possible:

- First and Foremost, My Lord and Saviour Jesus Christ, who was my source of strength and motivation throughout this thesis. You have made this project meaningful and showed me that your Grace is indeed sufficient for me. The Praise and the Glory belongs to you.

- My Beloved Wife (Trina), who was by my side through the high times and the low times. You where there to share my victories and tolerated me during my "perceived" defeats. You were a support to me even with your demanding schedule. You had confidence in me during times when I didn't have confidence in myself. I Love You.

- My Mother and Father (Mom and Dad), who were a constant source of encouragement and support. Thanks for your help along the way. You have been a blessing to me throughout my life.

- My Mother-in-law (Mother), who has done everything in her power to make these trying times as bearable as possible. I know that I promised to mention you in this thesis, but I would have done it anyway. You have been a great source of help and plus if it weren't for you, I wouldn't have such a beautiful wife.

- Professor Roger G. Mark (Dr. Mark), who was very patient with me throughout this project. You beared with me, even when it seemed you doubted whether I would complete this project (which made me more determined to finish). I have learned alot from you over these past few years that I will take with me throughout my professional career. I hope that one result of the time that we have spent together on this project is a lasting friendship.

- Dr. William J. Long (Bill), who gave so freely of his time. This project would not have been possible without your help. I am very grateful for the many hours that you devoted to this project. I will never forget you willingness to give.

- George B. Moody (George), who was my mentor in my early research days. You provided me with the foundation that I needed to complete this project. You bailed me out (Computers in Cardiology 1984) when I was caught in the middle of my transition to Medical School. You were always here (MIT), there (BIH) or somewhere (home, etc.) when I needed you. I appreciate all of the time that you devoted to this project, especially in the early stages.

- Gloria McAvenia (Gloria), who made appointments for me to see the chief, when there were no time slots available. You tolerated me when I would make life difficult by changing my appointments. You even put up with me when I would cancel my appointments altogether. You were a comfort during the times when I was to face the firing squad. Many Thanks.

- Dr. Paul Albrecht (Paul), who was a great help, especially when George was not here, there, or somewhere. I thank you for all of your help with the system at the BMEC, for your words of scientific wisdom - especially near the completion of your doctoral thesis when you seemed to be on another mental plane, and for serving on my panel of ECG Analysis Experts. I wish you the best in your future career endeavors.

- Professors Peter Szolovits (Pete) and Ramesh Patil (Ramesh), who provided me with the computer facilities and the technical advice that made this project a success. To Ramesh: I am going to beat you in "Horse" before I leave MIT. I would also like to thank all of the members of the Clinical Decision Making Group for their help along the way, and for tolerating my little pink "Please Do Not Disturb – Experiment in Progress" signs on the Symbolics Machines.

- Last, but not least, the National Consortium for Graduate Degrees for Minorities Inc. and the Rowland Foundation, who provided the generous financial support necessary for me to conduct this research. I will always be grateful for your committment to education and research.

Warren K. Muldrow

Contents

Chapter 1

Introduction

1.1 Background Information

Long-term ambulatory monitoring of the electrocardiogram has become an important diagnostic tool for physicians. Patients with various cardiac maladies who benefit from such an evaluation include [1,2,3,4,5,6]:

1. Individuals with known ventricular ectopic activity.

2. Those with previous myocardial infarctions.

3. Those with intermittent symptoms possibly related to arrhythmias or ischemia.

4. Those receiving antiarrhythmic drug therapy.

5. Individuals requiring long-term pacemaker evaluation.

There are two basic methods for analyzing the ECG during long-term ambulatory monitoring [7]. The first method involves continuously recording the ECG over a 24-48 hour period. The recording is subsequently analyzed by a trained technician using a high speed (60-480X real time) tape scanning system. A disadvantage of this approach is that the scanning process takes approximately an hour

of technician time. The second method is to process the ECG data in real time, an option made possible by microprocessor technology. Most of these systems store only the ECG segments that contain significant arrhythmic events. The major disadvantage of this type of system is that a large portion of the ECG data is lost, making it impossible to correct false negative errors.

Another very important application of real-time arrhythmia detectors is in coronary care units (CCU). In this clinical setting, significant or possibly life-threatening arrhythmic events must be detected and the appropriate alarms activated in order for patients to receive prompt medical intervention.

Although automated ECG arrhythmia detection has improved considerably over the past 15-20 years, noise and artifact are still a significant problem. Of foremost concern is noise due to electrode motion that contains strong spectral components in the range of the ECG frequency band. This type of artifact tends to cause false positive QRS detections and may trigger false alarms. Muscle artifact is also a problem, particularly at higher noise to signal ratios.

Thus, an important requirement for real-time arrhythmia detectors is sufficient noise rejection. Conventional arrhythmia detectors employ digital filtering techniques (eg., matched filters) to attain a level of noise rejection compatible with clinical application. Yet, many algorithms fall far short of the performance attainable by a "human expert". This fact was substantiated by comparing the noise performance of a state of the art real-time arrhythmia detection algorithm (ARISTOTLE [8,9] - Developed in the Biomedical Engineering Center, MIT) with the performance of a human expert. [1]

A Noise Stress Test [10] was used to superimpose electrode motion and muscle artifact noise onto an American Heart Association database tape [11,12] (Series 4001). The noise was added at various noise to signal ratios and presented to both ARISTOTLE and various human experts. As illustrated in Figure 1.1, the

[1] The Human Experts used in this study represented individuals with considerable experience in analyzing noisy ECG's. Their fields ranged from cardiologist, to research scientist, to system developer.

QRS sensitivity of ARISTOTLE in the presence of electrode motion noise was comparable to that of the human expert. Yet, the human expert far outperformed the algorithm in terms of the QRS and PVC positive predictivity and the PVC sensitivity. In terms of the PVC sensitivity, which is an important measure of performance indicating the likelihood that an arrhythmic event will be detected, ARISTOTLE's performance was 30% at a noise level of 0.4. [2] The human expert did not show signs of failure until subjected to a noise level of 0.9, where the PVC sensitivity fell just below 90%. The human expert also far outperformed ARISTOTLE with muscle artifact noise superimposed on the ECG. Examples of the ECG with various levels of electrode motion noise superimposed are presented in Figure 1.2.

These preliminary experiments revealed that human experts resort to knowledge of the underlying rhythm for a given patient, recognition of previous beat patterns, timing information (eg., average R-R interval and PVC coupling interval), and basic rules for cardiac rhythm disturbances while analyzing noisy ECG recordings. The morphology of the beats was of secondary importance in the presence of noise.

In order to determine whether the timing information was sufficient to accurately analyze noisy ECGs, the human experts were presented with ECG strips with most of the morphological information removed. These strips contained ARISTOTLE's beat annotations and bipolar spikes whose direction indicated detected peak polarities and whose magnitude was proportional to detected peak amplitudes (Figure 1.3). The experts were given a 30 second segment of "clean" (no noise) ECG where ARISTOTLE's beat annotations were correct. This was done in order for the expert to compile the appropriate knowledge base. The experts were instructed to correct the beat annotations of the contiguous 90 seconds, that contained a 60 second burst of electrode motion noise. The improvement in performance relative to ARISTOTLE operating alone is illustrated in Figure 1.4. There was as much as a 170% improvement in performance (PVC positive predictivity rose from

[2]The concept of the noise level is defined in chapter 3.

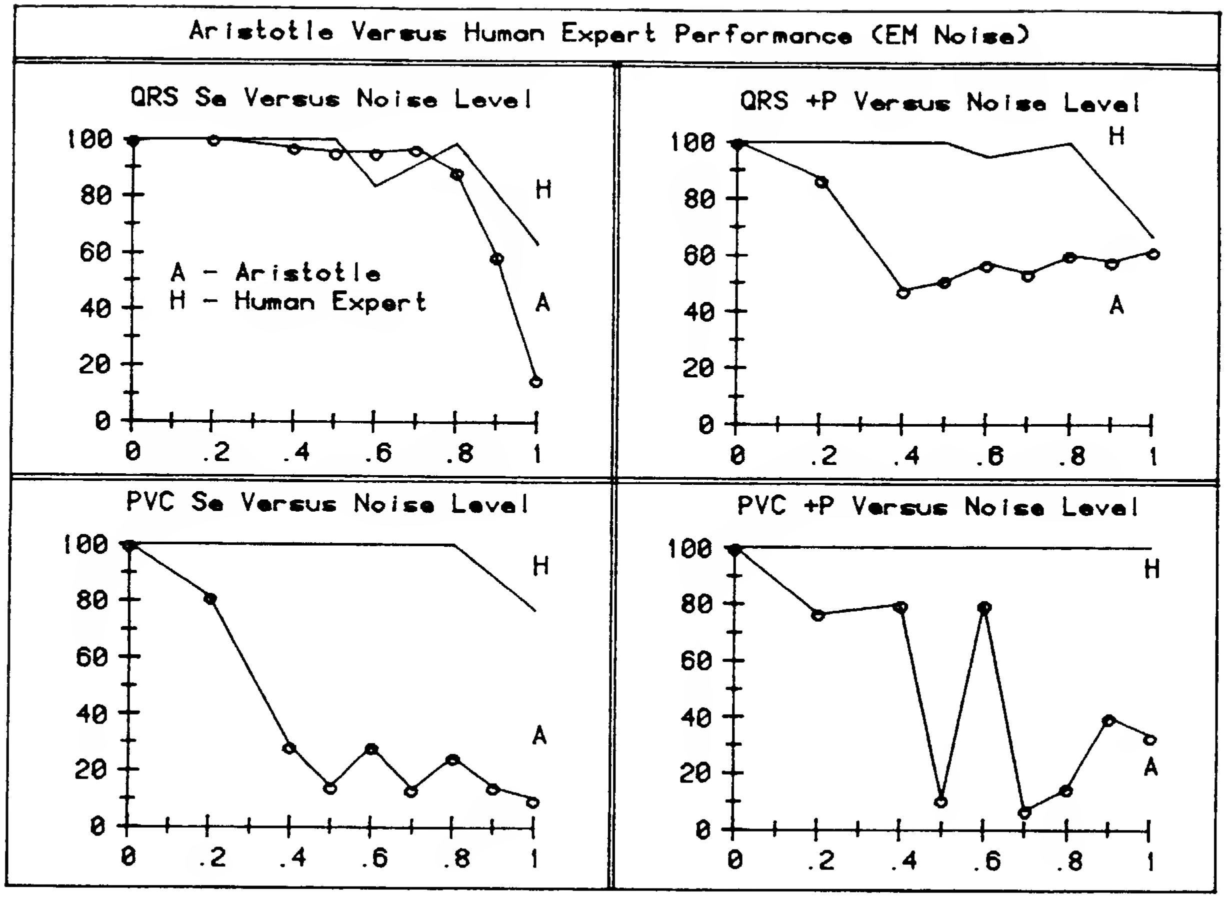


Figure 1.1: ARISTOTLE versus Human Expert Performance.

Plots labelled with *A* represent Aristotle's performance, while plots labelled with *H* represent the Human Expert. The chosen performance measures are the QRS and the PVC sensitivity and positive predictivity.

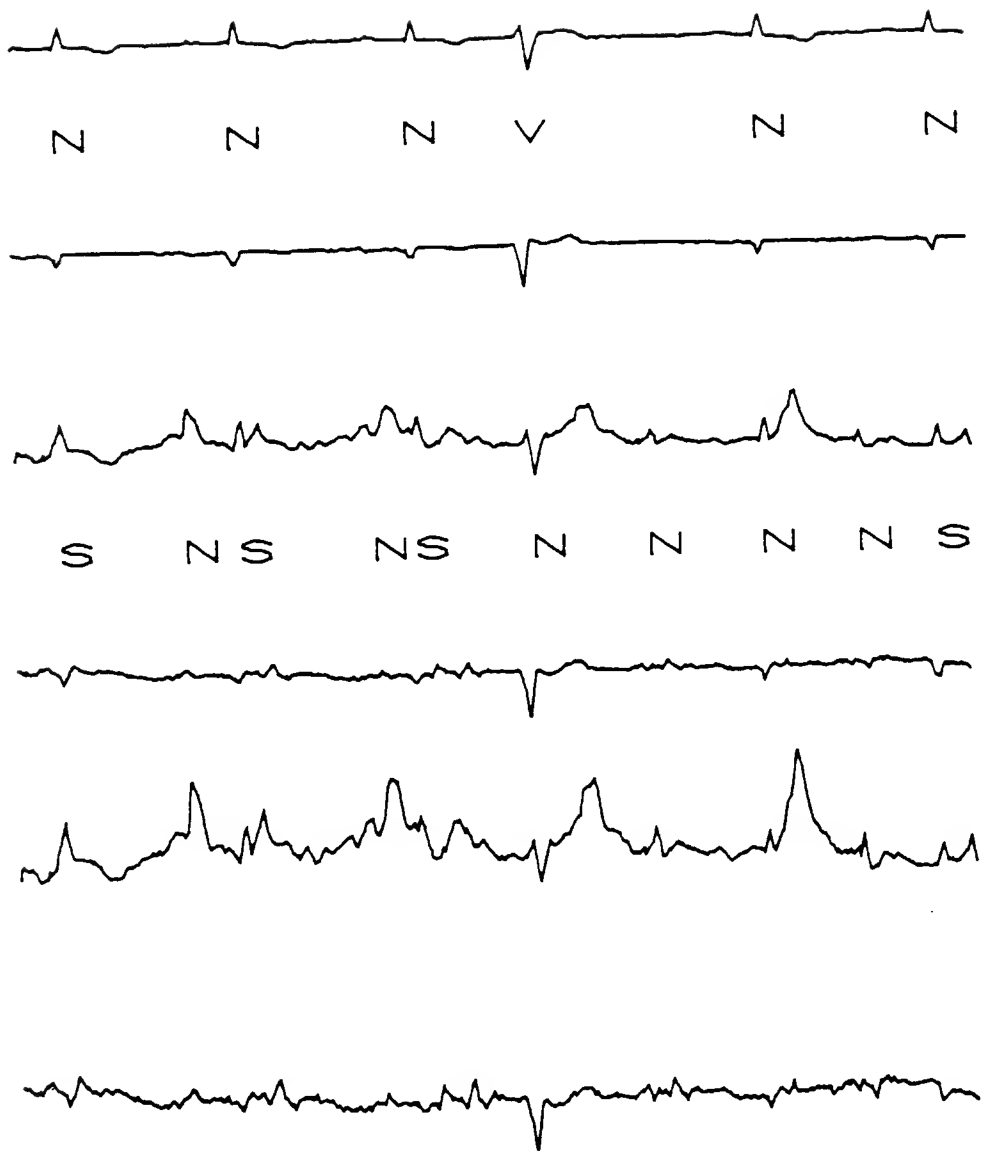


Figure 1.2: ECG with Superimposed Electrode Motion Noise.

A: AHA Database Tape 4001 (2 channels) without added noise. The truth annotations are shown between the 2 channels. N≡Normal, V≡VPB, S≡SVPB. B: Tape 4001 with electrode motion noise added at a level of 0.4. The annotations are those generated by ARISTOTLE while analyzing this noisy data segment. C: This represents Tape 4001 with electrode motion noise added at a level of 0.8.

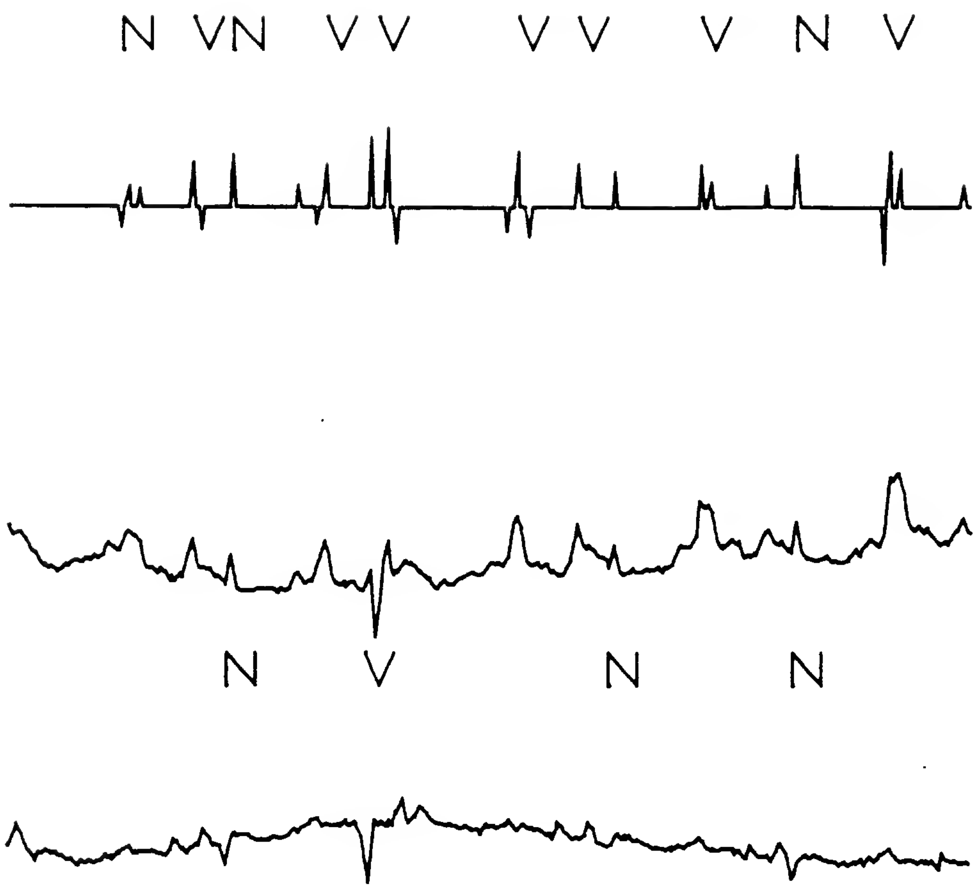


Figure 1.3: Data Presented to the Human Experts for Editing.

A: The top tracing represents the output of ARISTOTLE's matched filter. The beat annotations are those generated by ARISTOTLE. N≡Normal, V≡VPB. B: Represents the actual ECG segment that produced the matched filter pattern in A. The truth annotations are shown between the 2 ECG channels.

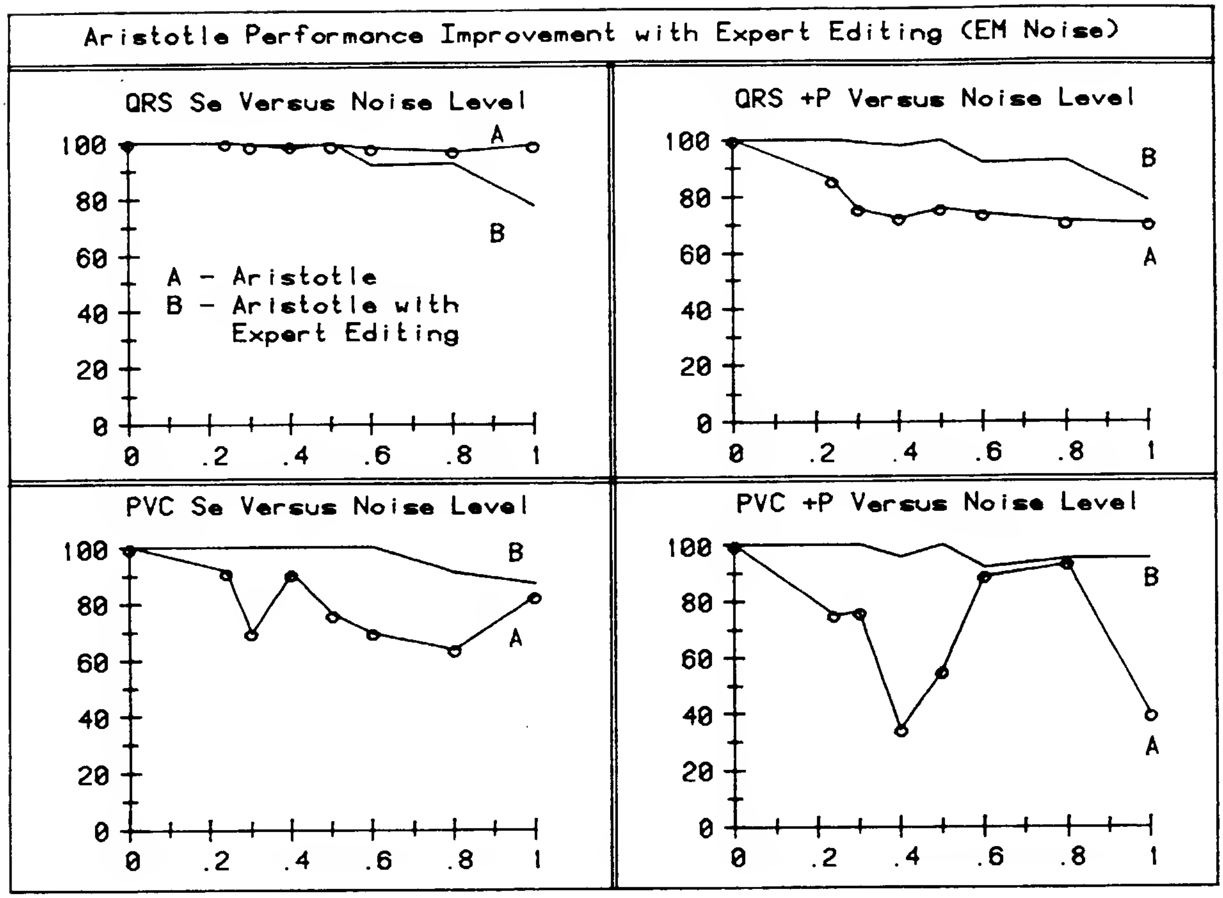


Figure 1.4: ARISTOTLE's Performance with Human Expert Assistance.

Plots labelled with *A* represent ARISTOTLE's performance while processing the noisy ECG segment. The plots labelled with *B* represent ARISTOTLE's performance with Human Expert assistance.

34% to 95%) over the uncorrected algorithm output using this approach. These results show that the timing and limited morphological information contained in ARISTOTLE's data stream are not optimally exploited during noisy ECGs.

1.2 CALVIN: A Novel Approach

CALVIN (CALipers in Very Intense Noise) represents an attempt to model the human expert approach to analyzing noisy ECG data. The system, which operates in series with ARISTOTLE, assists in the classification of beats during noisy segments of ECG by utilizing the knowledge base and the protocol of the human expert. The philosophy of the system is that the analysis of noisy ECGs can be improved by utilizing the information acquired from prior noise-free data segments. This information includes the characteristic timing (eg., R-R intervals and VPB coupling intervals) and the underlying rhythm for a given ECG.

When the noise level is low, the human expert is able to classify beats based primarily on visual inspection of the morphology. Under these low noise conditions, CALVIN accepts ARISTOTLE's beat classifications. ARISTOTLE behaves like a morphology driven algorithm (i.e., the classification of beats is based heavily on beat morphology).

When the noise level rises significantly, it becomes difficult for the human expert to visually distinguish true QRS complexes from noise artifact. This is because the noise, especially electrode motion noise, has strong spectral components in the ECG frequency band. They must therefore rely more heavily on event timing and information previously acquired pertaining to the patient's underlying rhythm. By comparison, CALVIN ignores ARISTOTLE's beat classifications when the noise level rises above a predetermined threshold. The events are then classified by the application of the rules along with the information contained in the knowledge base.

When the noise level rises to extreme levels, the human expert skips those regions of the ECG where he is unable to distinguish the true QRS complexes, and

searches for a segment of data where he can resume accurate beat classification. Under these same conditions, CALVIN skips the segments of data that it is unable to process and searches ahead in the data stream for a segment where processing can resume.

The noise added to the ECG has strong spectral components in the ECG frequency band. The actual beats become distorted and visually indistinguishable from the noise glitches.

This thesis further describes the system architecture, the developmental process, the system evaluation, and the future prospects of CALVIN:

- Chapter 2 describes the overall system architecture and the interfaces between the various components.
- Chapter 3 provides an explanation of the need for and the development of the Noise Stress Test (NST). A sample NST of two versions of ARISTOTLE is included in this section.
- Chapter 4 describes in detail the system preprocessor (preCAL).
- Chapter 5 explains some of the basics of the YAPS production system, used to implement the human expert protocol. It describes the mechanics of how CALVIN's interfaces reformat the data to attain compatibility with this production system.
- Chapter 6 explains how the Human Expert protocol was extracted and implemented with YAPS. It reveals the basic approach of the Human Expert in analyzing noisy ECG's.
- Chapter 7 tells how the system was evaluated. The results are presented and discussed.
- Chapter 8 discusses the future of this project. Possible enhancements to CALVIN are presented.

Chapter 2

System Architecture

2.1 System Function and Interfaces

The overall system architecture is illustrated in Figure 2.1. ARISTOTLE processes the raw ECG data and creates an annotation file. The annotation file (Figure 2.2) contains a record for each detected event (including false positive detections) that includes the time of occurrence, an event classification (Normal, VPB, SVPB, etc.), a noise level estimate for both data channels, an indicator of the ECG channel being processed [1], and a morphology metric for both channels. The morphology metric used is the maximum output of a matched filter in ARISTOTLE's QRS detector. The annotation file is then processed by CALVIN.

CALVIN operates in three basic modes, Learn Mode, Assist Mode, and SITU Mode (Search Into The Unknown). During the Learn Mode, CALVIN generates a knowledge base that characterizes the ECG under analysis. The knowledge base consists of:

- The noise threshold to activate CALVIN.

[1] ARISTOTLE determines which of the 2 channels of input has the lowest noise level and processes that channel.

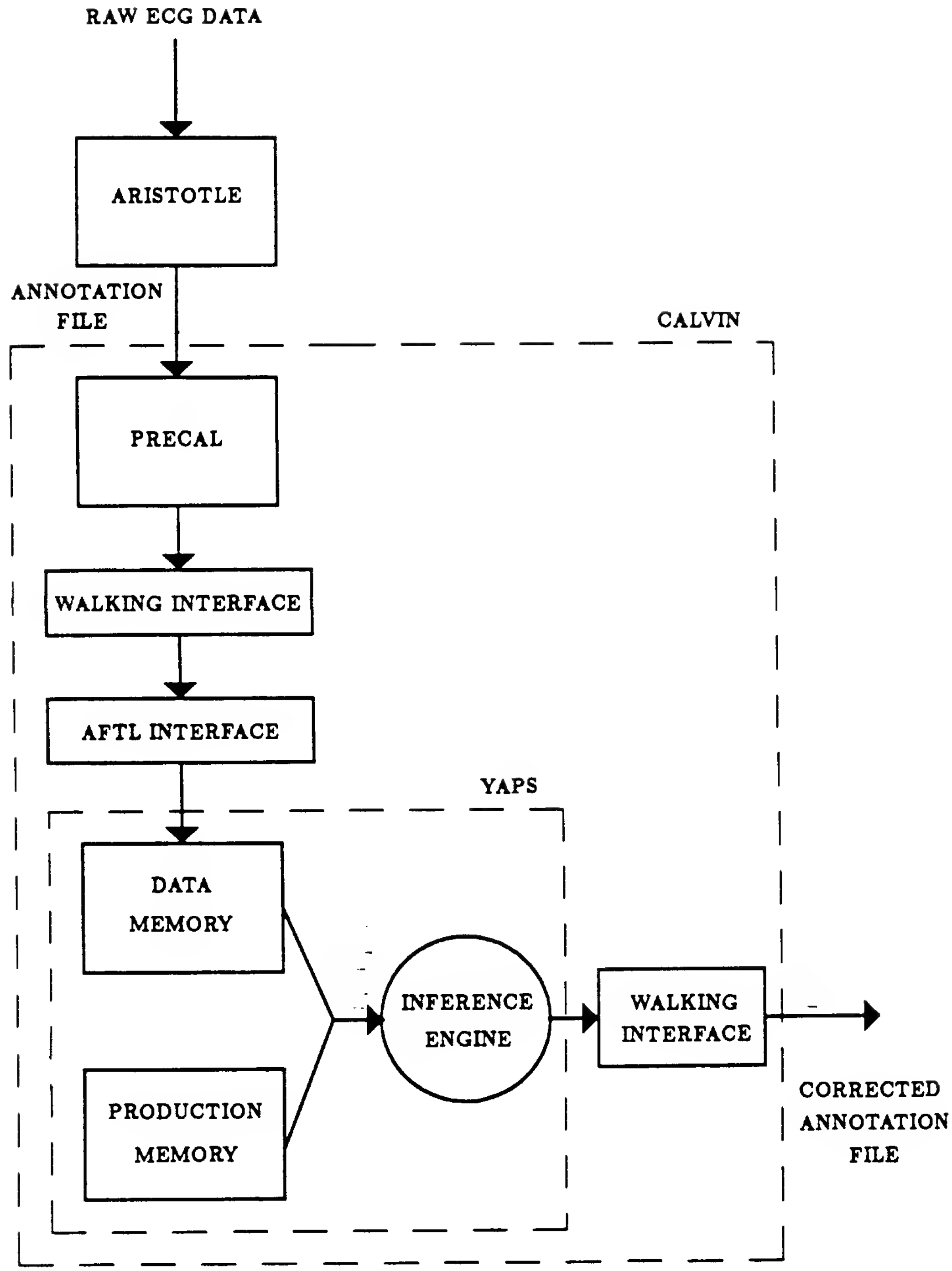


Figure 2.1: CALVIN System Architecture

8:00	120033	N	16	0	10	1100	1450
8:00	120103	N	16	0	9	947	1248
8:00	120199	N	15	0	8	983	1376
8:01	120375	N	14	0	7	1123	1437
8:01	120455	N	13	0	6	1048	1253
8:02	120592	N	12	0	5	1178	1352
8:02	120700	N	11	0	5	1097	1388
8:04	121009	N	10	1	4	1113	1328
8:04	121192	N	9	1	3	997	1302
8:05	121404	N	8	1	5	1037	1299

Figure 2.2: ARISTOTLE Annotation File

- The average and standard deviation of the matched filter output for Normal beats and VPBs.
- The N-N interval (time between two Normal beats) prediction.
- The N-V interval (time between a Normal beat and a VPB) average, standard deviation, and range.
- The V-V interval (time between two VPBs) average and standard deviation.
- The V-N interval (time between an isolated VPB and a Normal beat) prediction based on the N-N and the N-V interval information.
- The V-N interval (subsequent to a run of VPBs) average and standard deviation.
- The N-S interval (time between a Normal beat and an SVPB) average, standard deviation, and range, all normalized to the current heart rate.

- The S-N interval (time between an SVPB and a Normal beat) average, standard deviation, and range, all normalized to the current heart rate.
- The incidence of couplets, triplets, quadruplets, and ventricular tachycardia (VT – defined as greater than four successive VPBs). Also, the length of the longest run of VPBs.
- The length of the longest run of SVPBs.
- The percentage of VPBs that are interpolated and the percentage of premature beats that are SVPBs.

The preprocessor (preCAL), is responsible for compiling this knowledge base. During an initial 4 minute start-up period (and during all subsequent noise-free data segments), ARISTOTLE's beat classifications are assumed to be accurate. This assumption is reasonable, since one criteria for tape selection from the AHA Database for system evaluation was ARISTOTLE's near perfect performance on the tape in the absence of noise. [2] In a clinical setting, the noise level of the raw ECG data could be maintained at a low level and ARISTOTLE's performance could be evaluated during the learning period with the assistance of a technician and a temporarily immobile patient. PreCAL uses the beat classifications to generate a knowledge base that is representative of the ECG under analysis. Also, ARISTOTLE's noise level estimates are used to establish a threshold to distinguish "noisy" from "clean" ECG data. Once the 4 minute start-up period has elapsed, preCAL updates the knowledge base during all ECG segments with subthreshold noise levels.

PreCAL generates its own annotation file. This annotation file contains basically the same information as the one generated by ARISTOTLE with a few exceptions. First of all, the data channel under consideration by CALVIN is always channel 0. It does not currently have the capability to choose what it considers to be the best channel for analysis as does ARISTOTLE. Secondly, the system user

[2] The tape selection process for the evaluation of CALVIN is explained further in chapter 7.

has the option to choose the QRS morphology feature that will be included in the 7^{th} field of the annotation file and used internally by preCAL to characterize the beats on channel 0. Thirdly, this annotation file contains the knowledge base that has been compiled by preCAL, which is transferred at the beginning of each noisy ECG segment. Finally, since the beat classifications generated by ARISTOTLE are unreliable in the presence of noise above the threshold, they are all changed to Unknown (Q) by preCAL.

When the noise level of three consecutive beats exceeds the predetermined threshold, CALVIN exits Learn Mode and enters Assist Mode. PreCAL then transfers the current state of the knowledge base to its annotation file in the form of 10 entries with "NOTE" in the beat classification field. The knowledge base information is listed in the 7^{th} field, normally occupied by the beat morphology feature (Figure 2.3). During the Assist Mode, preCAL ceases to update the knowledge base. The preCAL annotation file is reformatted by the AFTL Interface into a form compatible with YAPS. The data is then loaded into Data Memory. The human expert protocol residing in Production Memory, along with the knowledge base are applied by the Inference Engine to the events detected by ARISTOTLE to select and reclassify the true QRS complexes. This information generated by CALVIN is used to create a modified and more accurate annotation file.

When the noise level rises to an extreme level where CALVIN is unable to process the event sequence, the system enters SITU Mode. During SITU Mode, CALVIN "shuts down" and searches ahead in the data stream for a segment where the event sequence matches one of several previously defined templates. When a match is located, beat classification resumes.

Once the noise level remains below the threshold for 20 consecutive beats, CALVIN reenters the Learn Mode and the updating of the knowledge base is resumed. The 20 beat lag was provided to give ARISTOTLE a chance to stabilize after exposure to noisy ECG data.

8:57	134362	N	1	0	3	48
8:57	134462	V	1	0	2	360
8:58	134740	N	0	0	1	-428
8:59	134920	Q	3	0	4	-345
8:59	134921	NOTE	3	0	4	nthresh 4
8:59	134922	NOTE	3	0	4	nn 180.1 6.0 18.0 27.0
8:59	134923	NOTE	3	0	4	nv 103.3 6.0 17.9 29.9
8:59	134924	NOTE	3	0	4	nvmaxmin 120 96
8:59	134925	NOTE	3	0	4	vn 256.9 13.4 40.2 67.0
8:59	134926	NOTE	3	0	4	ns 0.0 0.0 0.0 0.0
8:59	134927	NOTE	3	0	4	nsmaxmin 0.0 0.0
8:59	134928	NOTE	3	0	4	sn 0.0 0.0 0.0 0.0
8:59	134929	NOTE	3	0	4	snmaxmin 0.0 0.0
8:59	134930	NOTE	3	0	4	vvc 0.0 0.0 vvr 0.0 0.0 rvn 0.0 0.0
8:59	134931	NOTE	3	0	4	coup 0 trip 0 quad 0 vtach 0
8:59	134932	NOTE	3	0	4	interp 0.0 svpb 0.0 vrun 1 srun 0
8:59	134933	NOTE	3	0	4	qamp 2540.4 110.6 331.8 553.1
8:59	134934	NOTE	3	0	4	vamp 3013.7 80.5 241.5 402.5
9:00	135100	Q	5	0	5	104
9:01	135283	Q	5	0	7	-29
9:01	135380	Q	4	0	10	169
9:02	135613	Q	6	0	9	-1335

Figure 2.3: PreCAL Annotation File

2.2 Software Physical Location and Implementation

ARISTOTLE, preCAL and the AHA Database Tapes all reside on the PDP 11/44 at the Biomedical Engineering Center (BMEC) at MIT. ARISTOTLE and preCAL are implemented in the "C" programming language.

The human expert protocol is implemented on a modified version of the YAPS Production System. The AFTL Interface is implemented in Zeta Lisp. Both reside on Zermatt (Symbolics 3650) in the Laboratory for Computer Science (LCS) at MIT and are run on the Symbolics 3600 series Lisp Machines (Release 6 Operating System). [3]

The physical location of the various software components requires that the raw ECG data be procesed by ARISTOTLE and preprocessed by preCAL at the BMEC. The annotation file generated by preCAL is written onto magnetic tape and transferred to the LCS. The data is reformatted by the AFTL Interface and processed by CALVIN. It is once again written onto magnetic tape and transferred to the BMEC, where the results are used to generate the modified annotation file. This process of writing information onto magnetic tape and transferring it between the LCS and the BMEC comprises the Walking Interface (see Figure 2.1). A User's Manual for CALVIN is under separate cover.

[3] These components of CALVIN should *theoretically* run with Release 7. It has not been attempted and no guarantees are given.

Chapter 3

The Noise Stress Test

3.1 Background

An important aspect of cardiac arrhythmia detector performance is noise rejection. False positive QRS detections resulting from inadequate noise tolerance are one of the most important problems in a clinical setting. Conventional ECG database recordings, in general, do not exhibit the variety and intensity of noise necessary to sufficiently stress the noise rejection logic of an algorithm. This is especially true for the AHA Database, which tends to be relatively noise-free. Algorithm evaluations with a conventional ECG database alone would leave the developer with an inaccurate estimate of a detector's performance in a clinical environment.

Developers of arrhythmia detectors require a test of noise rejection that is both quantitative and reproducible. Reproducibility is especially important during the algorithm optimization process. This allows the developer to determine the effect of modifications on detector performance. Conducting noise performance evaluations in a quantitative (and standardized) manner is useful in the comparison of two different algorithms, especially those developed at independent sites.

Investigators have proposed various techniques to test the noise rejection logic of arrhythmia detectors [10,13,14]. The methods used to generate the noise include

the use of digital and analog simulators, the filtering of the ECG from noisy ECG recordings, and the recording of noise from electrodes placed on a subject in a configuration that excludes the ECG from the signal. There are drawbacks to each of these methods. First of all, one cannot be sure that the noise generated in an artificial manner has the same characteristics as that observed in a clinical setting. This factor applies to the specially configured electrode method, but is especially relevant to the noise simulation technique. Secondly, it is not a trivial matter to completely remove the ECG signal from a noisy recording.

3.2 Development of the Noise Stress Test

With the above criteria and drawbacks in mind, it was felt that the specially configured electrode method provided the best combination of both ease of implementation and real-world noise characteristics. A Noise Stress Test (NST) was developed using this method, that will more accurately predict a cardiac arrhythmia detector's performance under noisy conditions.

Volunteers were chosen to wear the Avionics model 445 Holter recorder, with electrodes placed on their arms and thighs such that the ECG was not detected. The absense of the ECG was ascertained by observing the signal on a holter scanner. The subjects were engaged in vigorous physical activity. They also purposefully created electrode motion artifact by moving the electrodes. Thus, a rich and varied noise source was created. Approximately 25 hours of 2-channel noise recording were generated. The recordings were then digitized at 250 Hz using the same software routines and equipment used to generate the MIT-BIH ECG Database [15].

The noise on the tapes was sorted into three major categories (baseline wander, electrode motion, and muscle artifact) on the basis of visual inspection. Baseline wander is a low frequency noise generated by movement of the patient and the relative position of an electrode pair. A representative sample of noise visually catagorized as baseline wander is shown in Figure 3.1. The second noise class is

Figure 3.1: Baseline Wander Noise Sample

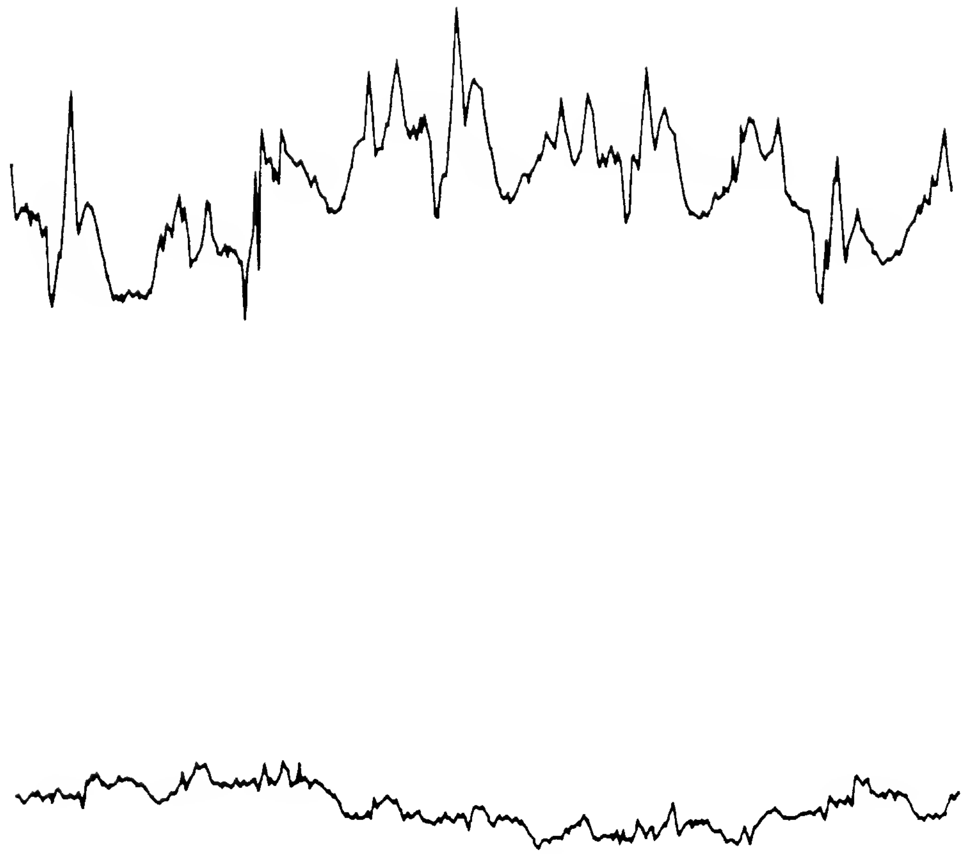


Figure 3.2: Electrode Motion Noise Sample

electrode motion. This type of noise has strong spectral components in the ECG band, making it the greatest problem for many ECG analysis programs. It is generated by transient mechanical forces acting on the electrodes. An example of the noise visually catagorized as electrode motion is presented in Figure 3.2. The final noise type is muscle artifact, generated by the electrical activity of skeletal muscle groups in the proximity of the electrodes. This class of noise is of a high frequency and is illustrated in Figure 3.3.

The 25 hours of data were scanned visually and segments were identified that distinctly represented each of the noise classes. It was difficult to find segments of electrode motion and muscle artifact noise without an underlying low frequency

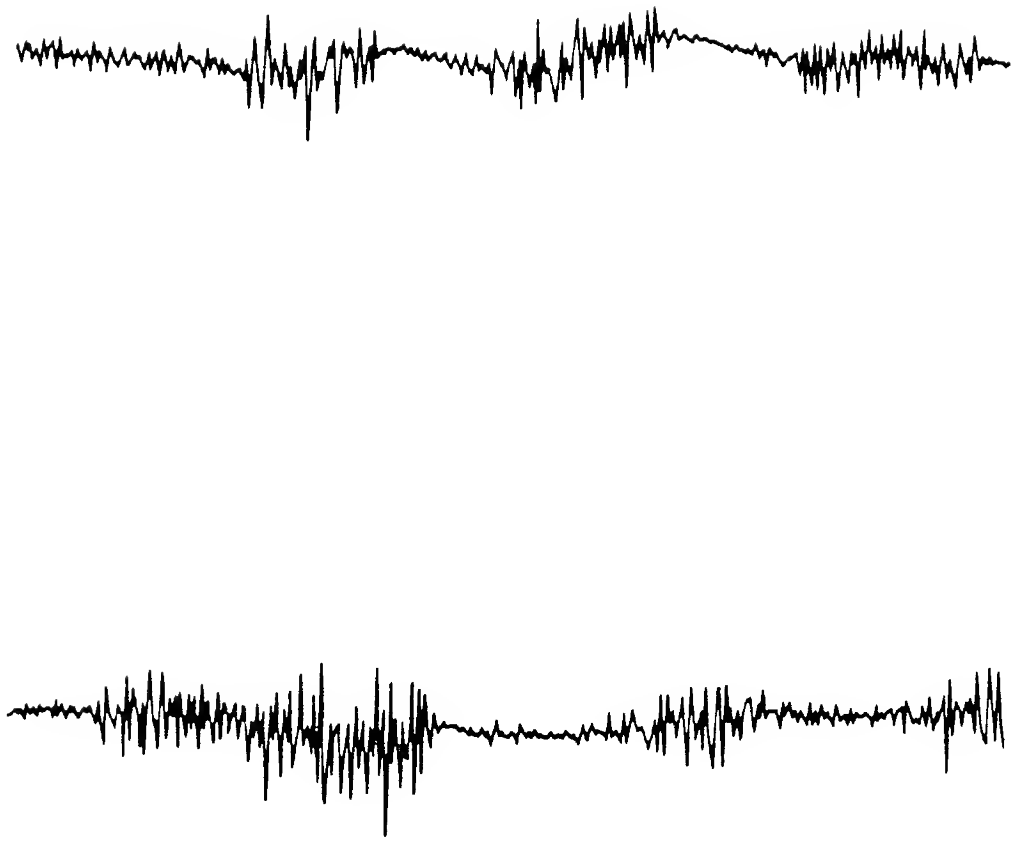


Figure 3.3: Muscle Artifact Noise Sample

component. Therefore this combination of noise types was considered to be acceptable. The segments of each noise type were sorted, then they were concatenated in a random order to generate three continuous 35 minute recordings. Discontinuities in the signal were avoided by matching the slopes at the opposing ends of the noise segments. The 16 bit format for the data was used rather than the 8 bit first difference format to avoid slew rate errors, especially with the higher frequency noise [16]. Access to these noise tapes is identical to that for the ECG database tapes (ie. via the EKG Database Utility Programs).

The test protocol for the NST is illustrated in Figure 3.4. Noise is added to the ECG tape on a sample-by-sample basis for the two channels. It is then input to the algorithm being tested. There is an adjustable gain factor for each of the noise channels, controlled by a noise protocol file (described in the CALVIN User's Guide). The ECG analysis program generates an annotation file that is compared to the truth annotation file for the chosen ECG tape to generate the beat-by-beat performance statistics, the QRS and PVC sensitivity and positive predictivity. [1]

Before a meaningful NST could be performed on an algorithm, the noise data had to be quantified and normalized to the ECG tape being used. A direct measurement of the average amplitude of the noise data would overestimate the value, due to the significant low frequency component on each of the noise tapes. Therefore, the noise tapes were divided into 10 second segments [2], the D.C. components were removed, and RMS amplitude was computed for each of the segments. The average RMS amplitude over all of the 10 second segments was then computed for each noise type. The ECG tapes were characterized by determining the average peak-to-peak amplitude of the normal QRS complexes, excluding outliers. [3]

[1] The compilation of the performance statistics is performed using the **compare** and the **stats** routines described in the EKG Database Applications Programs Manual. The use of these routines is also explained in the CALVIN User's Guide.

[2] This time interval was chosen because it is short relative to the period of the noise low frequency component.

[3] The average peak-to-peak amplitude of the ECG tapes was determined using the feature files generated by ARISTOTLE and the **feathist** routine also developed by George B. Moody, Biomedical Engineering Center, MIT.

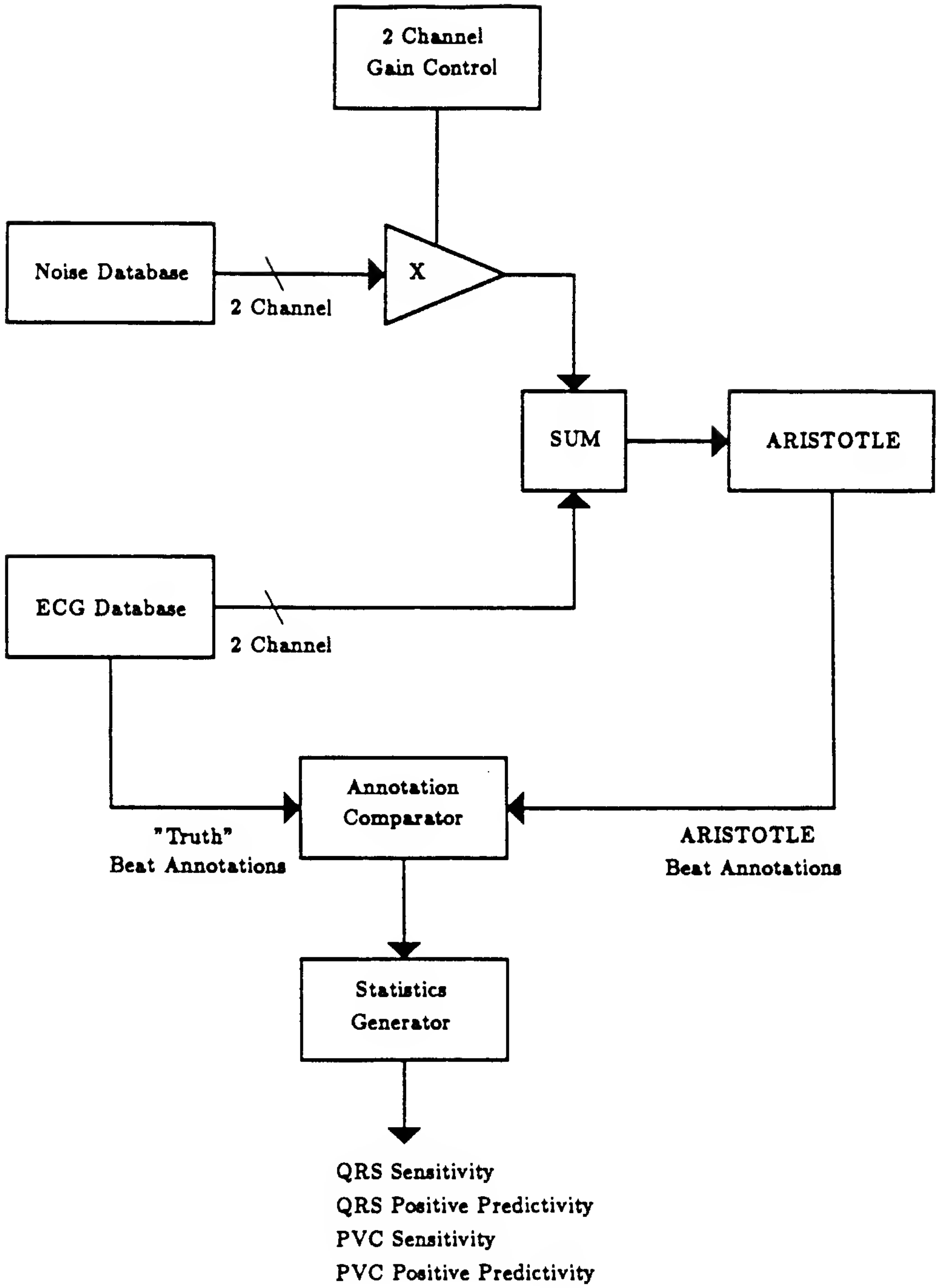


Figure 3.4: Noise Stress Test Protocol

Noise Type	RMS Amplitude Channel 0	RMS Amplitude Channel 1	RMS Amplitude Ratio Channel 0 / Channel 1
Baseline Wander	67.95	28.41	2.392
Electrode Motion	133.9	38.09	3.515
Muscle Artifact	38.63	35.45	1.090

Table 3.1: RMS Amplitude Calculation Results for the 3 Noise Types

The noise was then added to the ECG at gain settings such that the noise-to-signal amplitude ratio was equal on the two channels. The Average RMS amplitudes for the three noise types are presented in Table 3.1. It should be noted that these values are not in volts, but in ADC units. Instead of presenting the data in terms of the noise-to-signal ratio, a value referred to as the "noise level" is used. This unit of measure, defined in Equations 3.1 and 3.2, was used because the noise was uncalibrated in terms of ADC units/mV.

$$Noise\ Level\ Channel\ 0 = Gain_0 \times \frac{10^2}{PP_0} \tag{3.1}$$

$$Noise\ Level\ Channel\ 1 = Gain_1 \times \frac{10^2}{PP_1} \tag{3.2}$$

$$Where\ PP_i \equiv the\ peak-to-peak\ amplitude\ of\ the\ i^{th}\ ECG\ channel$$

The gain of channel 1 is adjusted (multiplied) by one of the ratios in the 4^{th} column of Table 3.1, depending upon the noise type in use. This takes into account the disparity between the signal amplitudes on the two noise channels and assures that the relative noise-to-signal ratio is identical on both ECG channels.

3.3 Evaluation of Two Versions of Aristotle

In order to illustrate the use of the NST, two algorithms were tested. The first algorithm was ARISTOTLE, previously described in this document. The other al-

gorithm BAM (Bedside Arrhythmia Monitor) was developed in the Bioengineering Laboratory at the Beth Israel Hospital in Boston.

The protocol for the addition of noise consisted of an initial 5 minute learn period (noise-free region), followed by 4 minute bursts of noise separated by 1 minute noise-free segments. The Algorithms were stressed using both electrode motion and muscle noise. The noise level of the tests with the electrode motion noise ranged from 0.0 to 1.2 in increments of 0.2. The noise levels used in the tests with the muscle noise were 0.0, 0.1, 0.5, 1.0, 1.4, 1.8, and 2.0. Tape 4001 from the AHA Database was used for this series of tests.

3.4 Discussion of the Noise Stress Test Results

The results for this series of test are presented in Figures 3.5 - 3.8. Looking first at the results of the electrode motion stress tests, ARISTOTLE outperforms BAM in terms of the QRS Positive Predictivity, the PVC Sensitivity and the PVC Positive Predictivity (Figure 3.5B, 3.6A, 3.6B). Yet, ARISTOTLE appears to attain this superior performance at the expense of the QRS Sensitivity (Figure 3.5A). Looking at the actual number of detections made by both algorithms, it appears that ARISTOTLE was not active during some ECG segments at the higher noise levels. This accounts for the false negative detections responsible for the low QRS sensitivity.

The stress tests conducted with muscle noise show that BAM is the superior algorithm when exposed to this high frequency noise. ARISTOTLE's performance was comparable to BAM's in terms of the QRS sensitivity, yet BAM outperformed ARISTOTLE in terms of all the other measures (Figures 3.7 and 3.8). This fact may indicate that BAM has a better defined digital filter passband, especially the upper frequency limit.

Determining which algorithm is superior overall is a very subjective matter, dependent upon the particular application. A high QRS sensitivity is desirable in any

application, yet to accomplish this, there is a tradeoff with an increasing false positive detection rate. In an application such as a real-time holter monitor, a higher false positive rate may be acceptable in order to attain a higher QRS sensitivity. A high false positive detection rate in the CCU setting would be intolerable, since this would lead to a loss of user confidence and an increased response threshold of the CCU staff to any arrhythmia alarm, assumed to be a false alarm. Absolute superiority can be exhibited, as was shown in the case of the stress tests with the muscle noise. But when the results are equivocal, the proposed application must be considered.

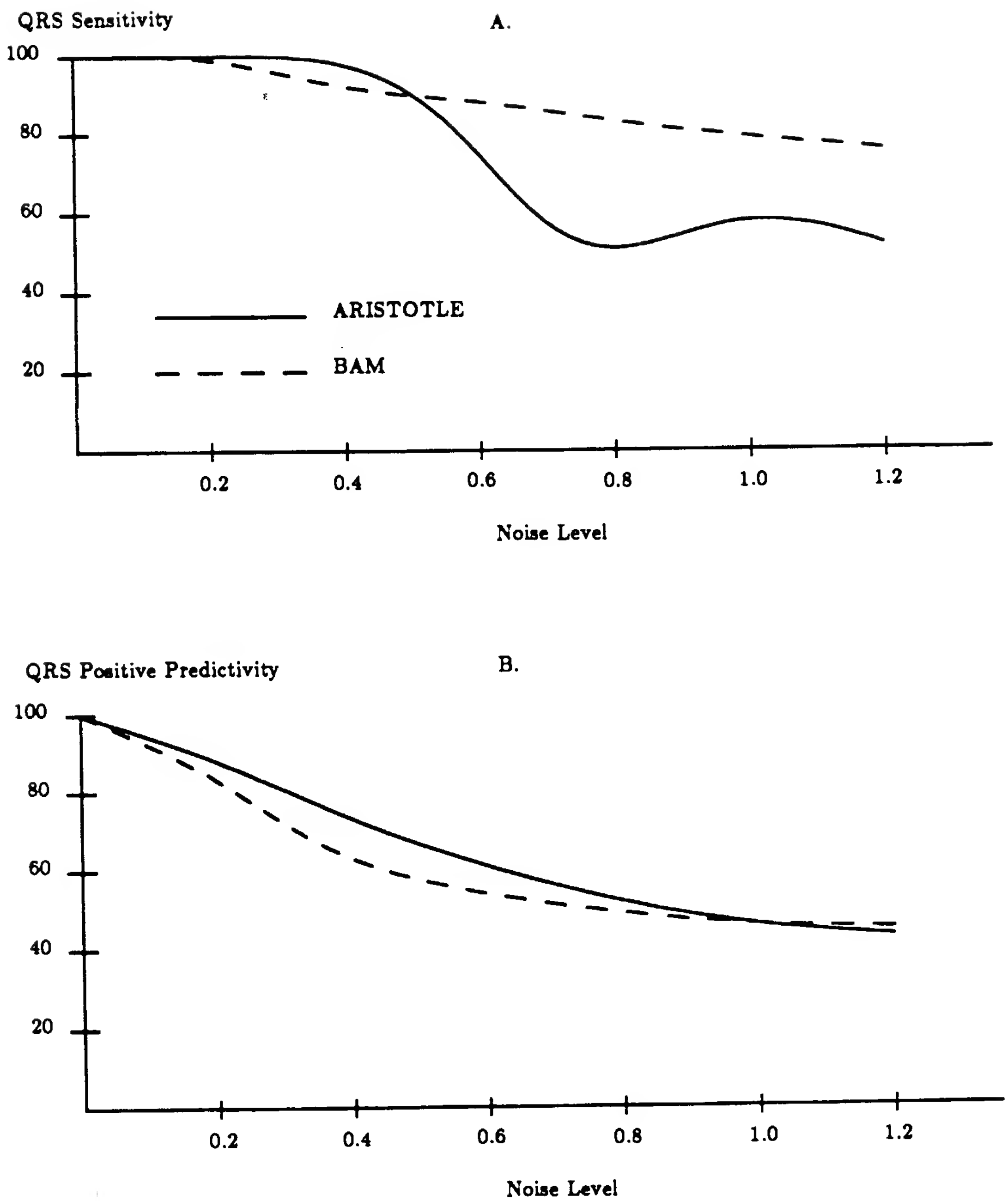


Figure 3.5: NST - QRS Performance with Added Electrode Motion Noise

These tests were conducted with AHA Database Tape 4001.

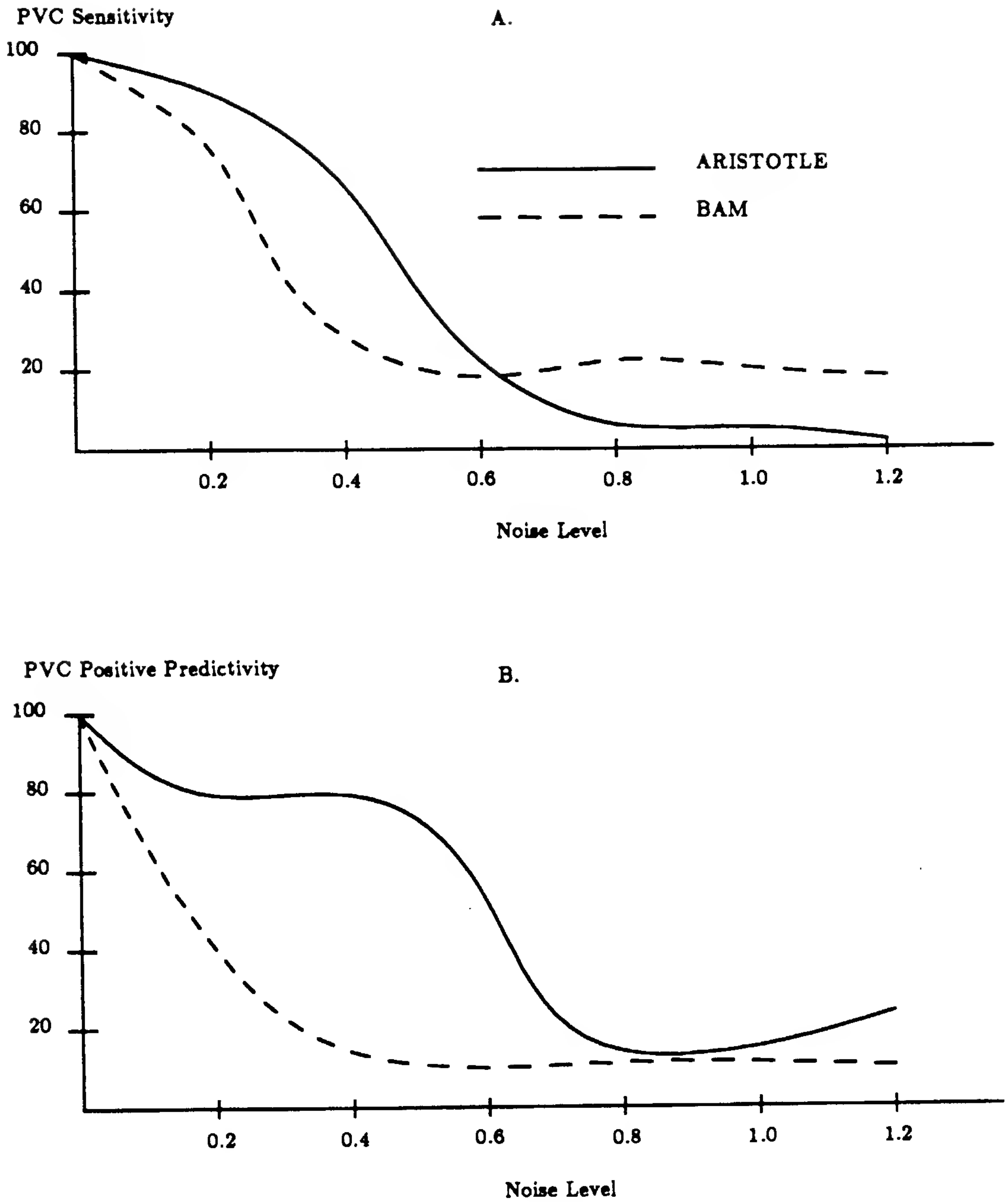


Figure 3.6: NST - PVC Performance with Added Electrode Motion Noise

These tests were conducted with AHA Database Tape 4001.

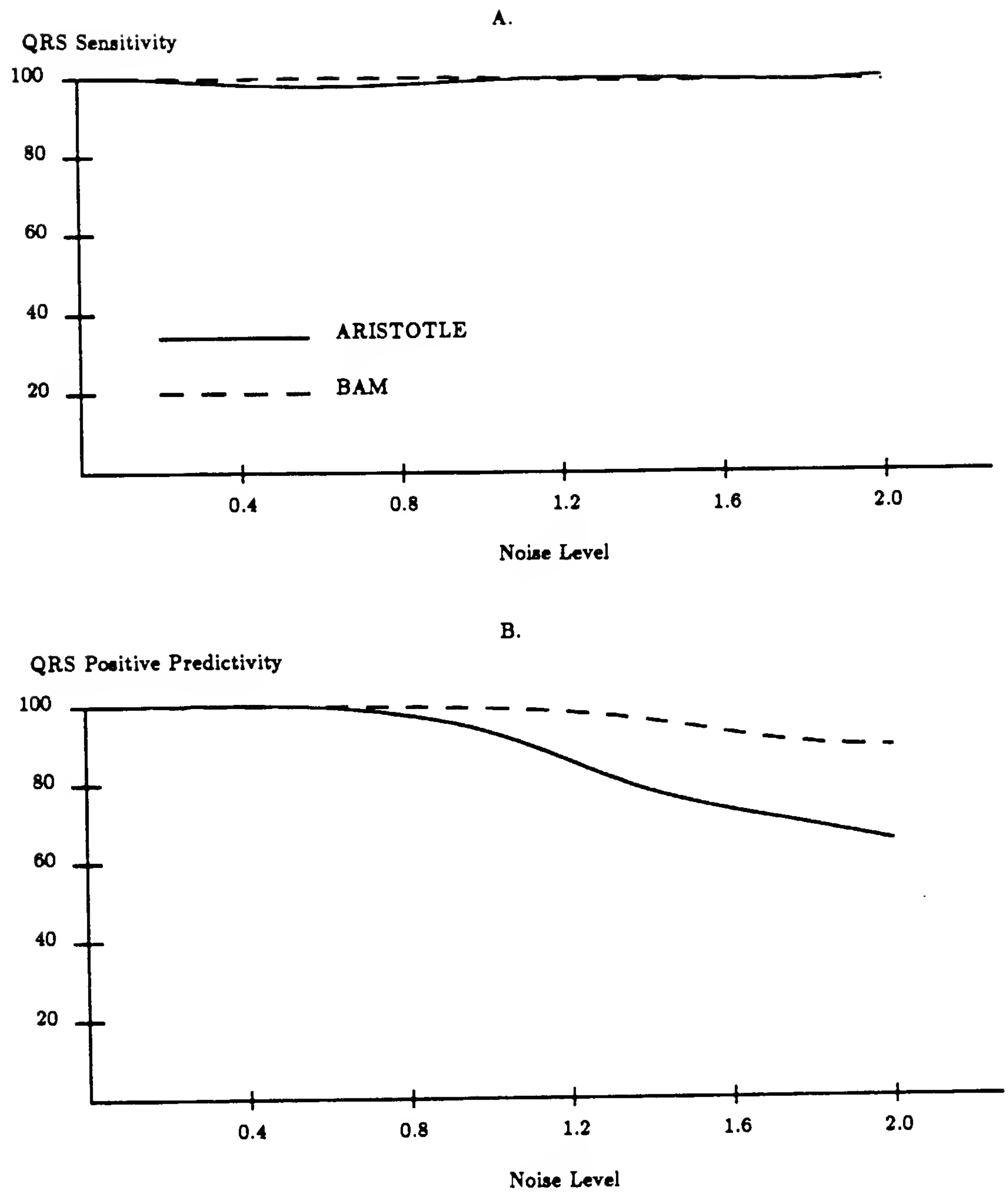


Figure 3.7: NST - QRS Performance with Added Muscle Noise

These tests were conducted with AHA Database Tape 4001.

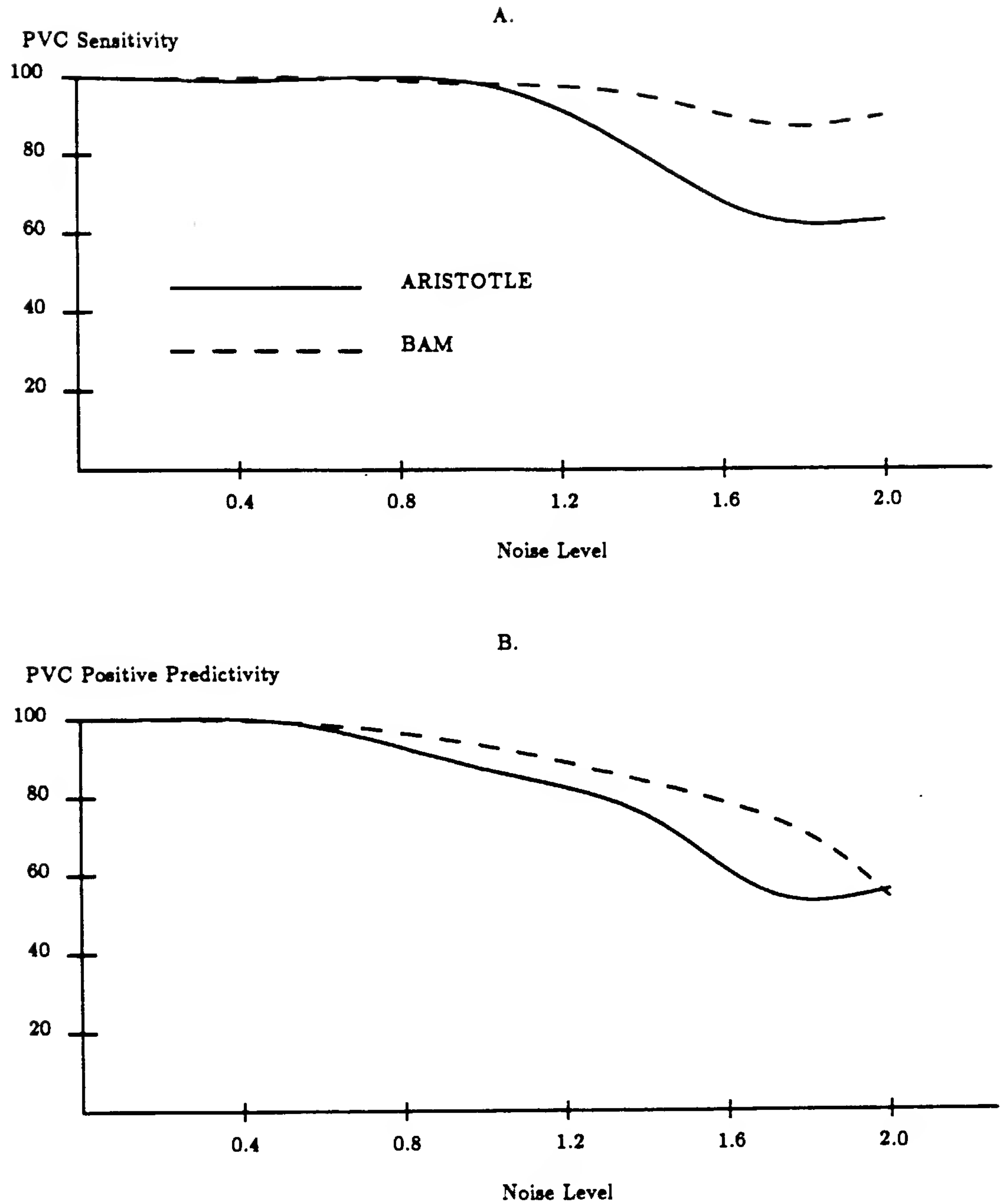


Figure 3.8: NST - PVC Performance with Added Muscle Noise

These tests were conducted with AHA Database Tape 4001.

Chapter 4

PreCAL: The System Preprocessor

PreCAL processes the annotation file generated by ARISTOTLE and compiles the knowledge base necessary for analyzing the noisy ECG data. In order to completely understand the software logic, one must become familiar with the Database Utility Programs, developed in the BMEC [17]. These utility programs serve as the direct interface between ARISTOTLE (the annotation file) and preCAL.

It should be mentioned at the outset that this description of the preprocessor refers to the most recent version to be used with the latest version of ARISTOTLE. The results of the system evaluation presented in chapter 7 correspond to an earlier version of both preCAL and ARISTOTLE. The rationale for describing the newest system is that future enhancements to CALVIN will use the latest version of ARISTOTLE, since it provides the greatest amount of flexibility. Any significant differences between the two versions of the preprocessor will be noted. The CALVIN User's Guide illustrates the use of both versions of preCAL.

4.1 PreCAL Architecture

The preprocessor architecture is illustrated in Figure 4.1. A beat record from ARISTOTLE's Annotation file and the corresponding feature file entry (which contains most of the morphological information for each beat) is input to preCAL. The Noise Detector then determines whether the beat information is to be processed by the Clean Data Processor (subthreshold beat noise level) or the Noisy Data Processor (suprathreshold beat noise level).

The Clean Data Processor (CDP) compiles the knowledge base during noise-free segments of ECG data. It utilizes ARISTOTLE's beat annotations (assumed to be accurate) to construct a knowledge base that is representative of the ECG under analysis. The Noisy Data Processor (NDP) operates on the noisy ECG segments. At the beginning of each noisy ECG segment, the current state of the knowledge base is transferred to preCAL's annotation file. All beat annotations are modified to include the noise level estimates and the beat morphology descriptor (the matched filter output for this study). The information for each processed beat is written to preCAL's annotation file. The preprocessor components are described in greater detail in the following sections.

4.2 The ARISTOTLE Feature File

In order to understand the flexibility provided by the latest version of ARISTOTLE, one must be aware of the information provided in the "feature file". This feature file, which is generated by ARISTOTLE, contains all of the data used by the algorithm to classify detected beats. [1]. Each feature file entry consists of a 36 byte block or 18 - 2 byte integers (Figure 4.2). There is a one-to-one correspondence between feature file entries and beat annotations.

The first item of each entry is the Data Channel. This item indicates which

[1] The older verson of ARISTOTLE does not generate an *accessible* feature file.

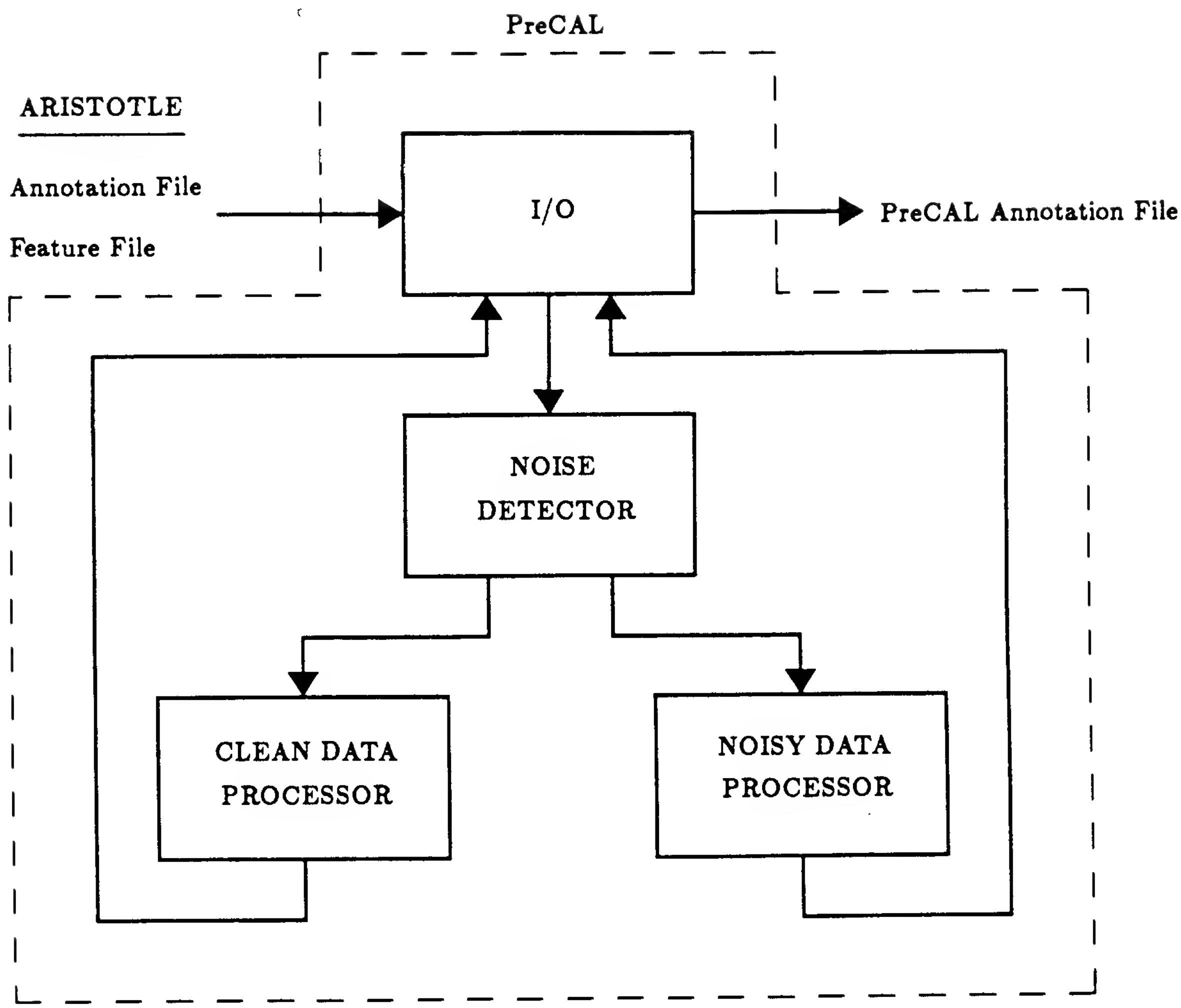


Figure 4.1: PreCAL Architecture

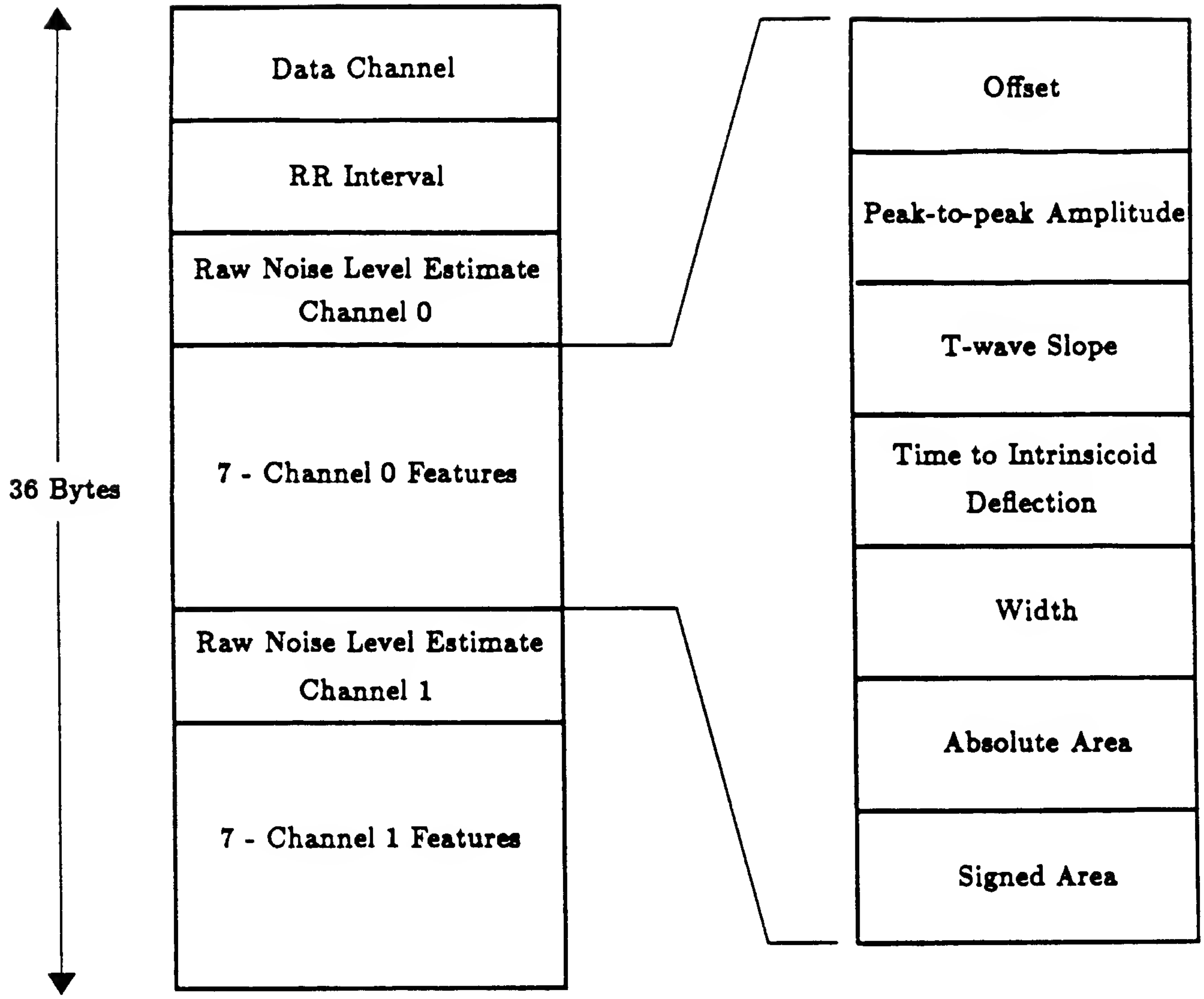


Figure 4.2: ARISTOTLE Feature File Data Structure

channel ARISTOTLE is processing for a given beat. A zero indicates channel 0, a one indicates channel 1, and a two indicates that both data channels are currently being processed (eg., during ARISTOTLE's learning period). The second item is the RR Interval, which gives the number of samples between the current and the previous beat. The Raw Noise Level Estimates are ARISTOTLE's estimate of the noise intensity in the region of the detected beat.

The 7 Features are computed by ARISTOTLE and used to classify the beat morphology. These consist of the offset, the peak to peak amplitude, the T wave slope, the time to intrinsicoid deflection, the beat width, the absolute beat area, and the signed beat area. Any one of these 7 features or some combination could be used to characterize the beat morphology for CALVIN [2].

There are three sets of criteria that must be satisfied by the chosen feature(s). First of all, the feature should be stable during noise-free segments of data. Secondly, the feature should exhibit resilience subsequent to a noisy segment of data. Finally, it is desirable to have a feature whose value for PVCs is only minimally altered during noisy data. [3] Experiments were conducted to identify which of the 7 features (and the matched filter output) best satisfied these criteria. Noise was added to an ECG recording in 4 minute bursts using the NST (Noise Stress Test) and the response of each morphology feature was observed. The single feature that best satisfied the above criteria was the matched filter output. The use of a multi-dimensional beat morphology descriptor (ie., more than one morphology feature) was not explored in this study.

PreCAL is currently able to handle any 1 of the 7 feature file morphology descriptors or the matched filter output. The noise level estimation and the morphology feature of the detected beats from both channels is extracted from the

[2] In the older version of ARISTOTLE, the matched filter output was the only morphological feature readily available. The matched filter output can also be generated for the latest edition of ARISTOTLE.

[3] This quality is desirable since it was noted during the analysis sessions with the human experts that PVCs were used as reference points in ECG segments with intense noise. The experts found it easier to distinguish PVCs from artifact than Normal beats. This was due to the usually larger amplitude of PVCs relative to Normal beats.

feature file entry, processed, and included in the preCAL annotation file. The noise level estimates for channel 0 and channel 1 are placed in field 4 and field 6 of the annotation file respectively. The morphology feature for channel 0 only is placed in field 7 of the annotation file. The subsequent processing of this information by preCAL is described below.

4.3 Noise Level Detector

The noise level estimate contained within the feature file is considered to be a "raw" estimate with a relatively unbounded range. This estimate is filtered (or "smoothed") in the noise detector in the following manner [4]. The smoothed noise level estimate is an integer in the range of 0-16. If the raw noise level estimate for the beat under analysis is greater than the smoothed estimate for the previous beat and less than or equal to 16, then the smoothed estimate for the current beat is set equal to the raw estimate. If the raw noise level estimate of the current beat is less than the previous smoothed noise level estimate, then the smoothed estimate is set to the previous smoothed estimate decremented by 1. If the current raw estimate is greater than 16, then the smoothed estimate is set equal to 16. This filtering routine, illustrated in Figure 4.3, allows a rapid rise, yet forces a gradual decline in the noise level estimate.

CALVIN operates on the assumption that ARISTOTLE's performance is error-free and the ECG is relatively noise-free during an initial 4 minute start-up period. A corollary of this assumption is that the average noise level observed during this period is compatible with accurate processing by ARISTOTLE. Therefore, the average noise level during the start-up period is computed for both of the ECG channels. The noise threshold is set to the largest average value plus 2 units. Reinitialization of the system is required to change the noise threshold after the start-up period has elapsed.

[4] The "smoothed" noise level estimate for both channels is included in field 4 and field 6 of the annotation file generated by the older version of ARISTOTLE.

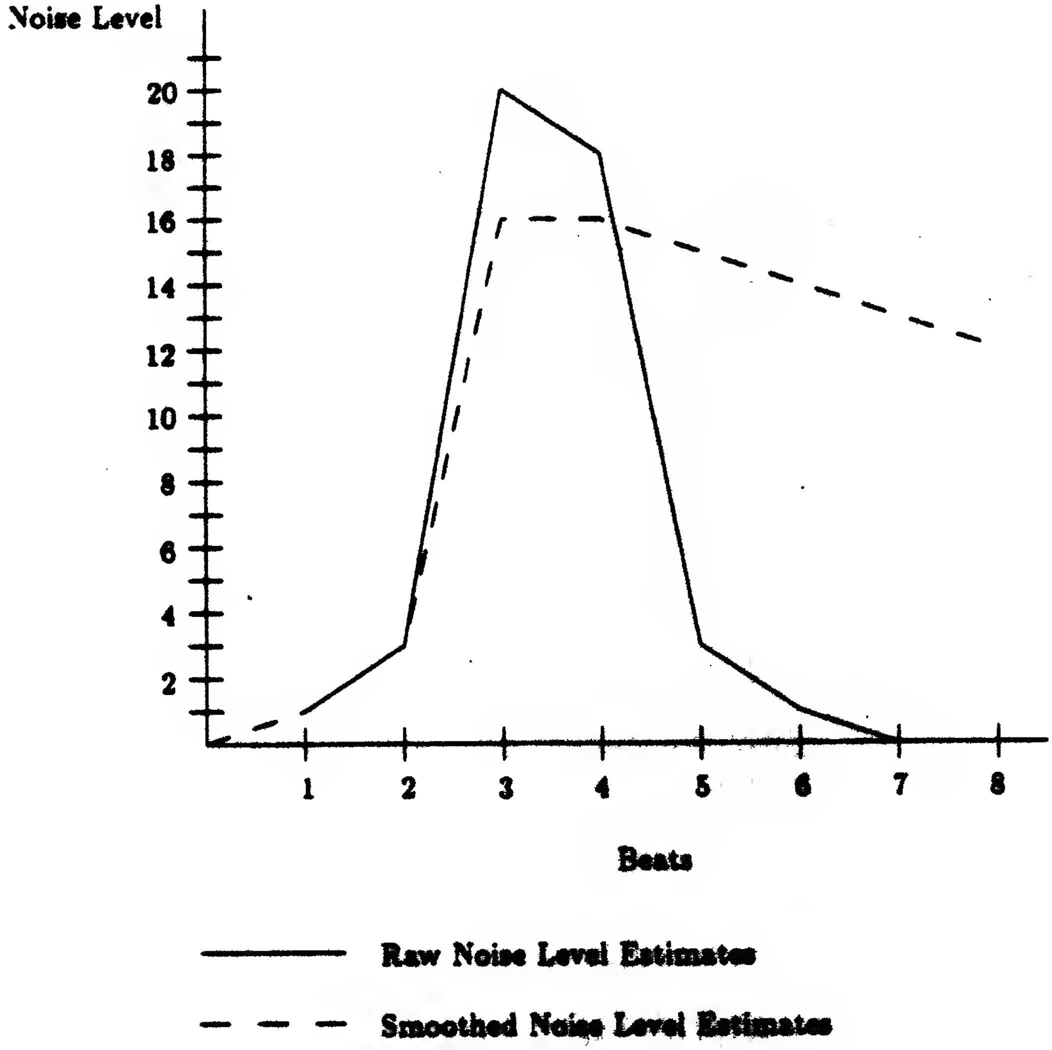


Figure 4.3: Filtering Routine Example

The start-up period serves another important function. It assures that a sufficient amount of data has been analyzed by CALVIN in order to generate a representative knowledge base prior to the processing of noisy ECG data.

PreCAL continually analyzes the noise level on both ECG channels. In order for CALVIN to pass from Learn Mode to Assist Mode, three consecutive beats on either ECG channel must have smoothed noise level estimates greater than the noise threshold. This method was used to avoid the activation of CALVIN during very transient increases in the noise level. Twenty consecutive beats with subthreshold noise levels are required to reenter Learn Mode. This lag period was provided to give ARISTOTLE time to stabilize after exposure to noisy ECG data.

4.4 Clean Data Processor

The CDP updates the ECG statistics during Learn Mode. Once the noise level exceeds the threshold, CALVIN shifts to the Assist Mode and the NDP is invoked. The statistics are no longer updated by preCAL, but are stored until the episode of noise has ceased. The current state of the knowledge base is transferred to the preCAL annotation file at the beginning of the noisy ECG segment. Once a subthreshold noise level persists for 20 beats, the CDP is again invoked and the updating of the knowledge base is resumed.

4.4.1 Modular Design of Clean Data Processor

The CDP consists of modules that operate on various sequences of beat classifications. During noise-free segments of data, the classification and the sample number of the fiducial point assigned by ARISTOTLE is stored for the the previous and the current beat. Also, the morphology feature for the current beat is extracted from the feature file. The classification of the previous and the current beat determines the module that is activated in the CDP. Once a CDP module is activated,

the appropriate knowledge base statistics are updated. PreCAL recognizes beat sequences involving Normal (N), VPB (V), SVPB (S), R on T PVC (R), Unknown or Unclassified (Q), and Learn (L - generated by ARISTOTLE during it's startup period) annotations used by ARISTOTLE. Currently, when preCAL does not recognize a beat sequence, it shuts down. This is not an ideal response for clinical applications, but proved to be very helpful during the system development.

In order to further illustrate the modular design of the CDP in preCAL, three specific modules will be described in the next few sections. Following this is a section providing the definition of the knowledge base parameters and an explanation of how the various entries are computed.

The Normal-PVC Module

The Normal-PVC Module of the CDP handles the N-V beat sequence. A flowchart of the functioning of this module is presented in Figure 4.4. Once the N-V beat sequence and the subthreshold noise level have been ascertained, the V morphology feature and the N-V coupling interval statistics are updated. Note that when the morphology feature statistics are updated, the VPB Counter is incremented. Also, the NV coupling interval statistics include the average, the standard deviation, and the range. The sample number of both beats and the classification of the current beat (V) are stored. The sample number of the Normal beat is stored so that the occurence of an interpolated PVC can be detected. The Premature Beat Counter is incremented and the SVPB/Total Premature Beats ratio is updated. [5] The beat information is then written to the preCAL annotation file.

[5] This ratio, where SVPB represents the total number of observed SVPBs and Total Premature Beats refers to the total number of observed premature beats, is used to estimate the probability that a given premature beat is an SVPB.

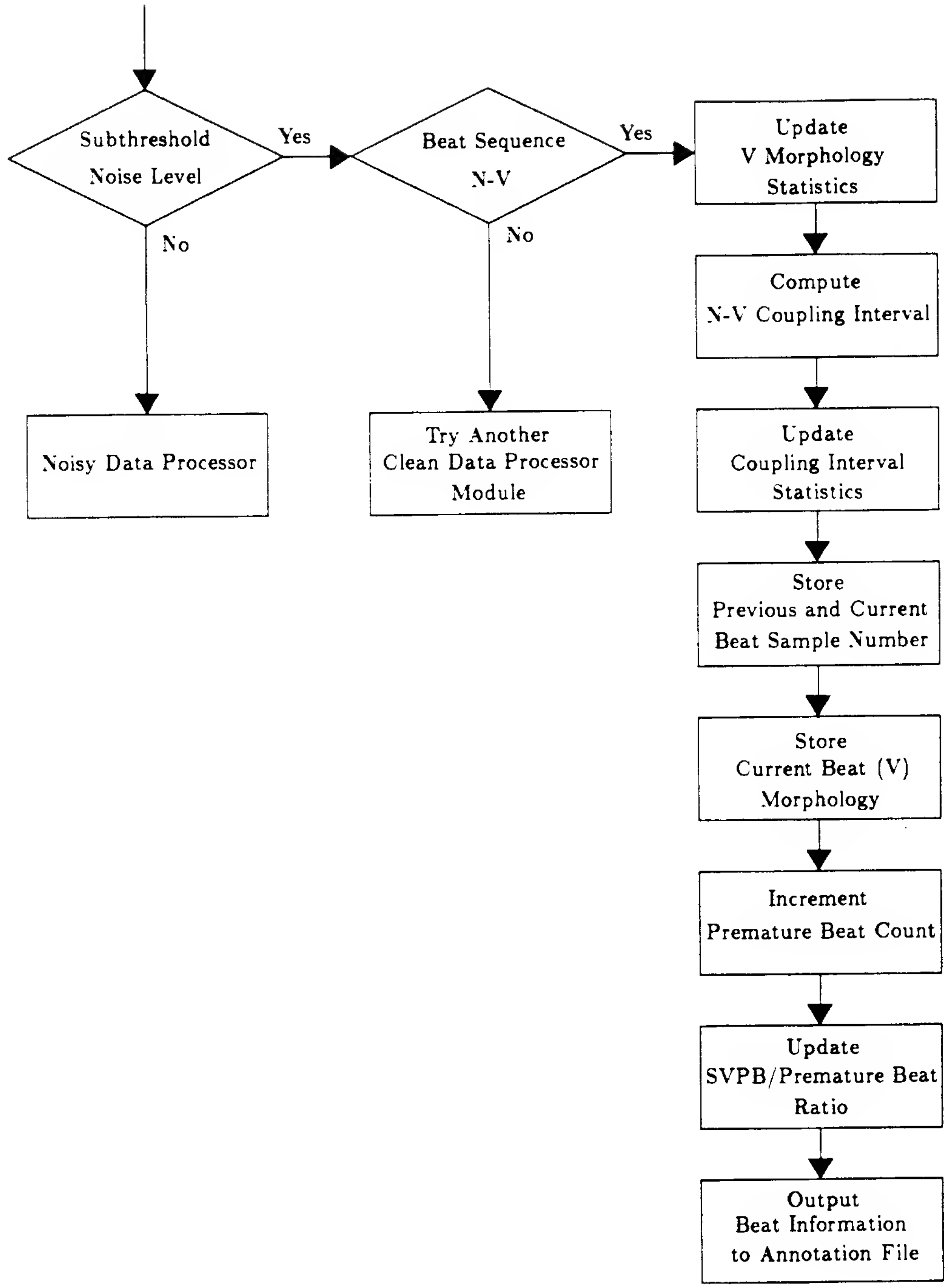


Figure 4.4: Normal-PVC Clean Data Processor Module Flowchart

The PVC-PVC Module

The PVC-PVC Module of the CDP handles the V-V beat sequence. A flowchart of the functioning of this module is presented in Figure 4.5. The appropriate beat sequence and a subthreshold noise level activates this module. A run counter that keeps track of the number of successive PVCs is incremented. Then the V morphology statistics (including the VPB count) are updated. The V-V coupling interval is computed and the appropriate statistic is updated depending upon the length of the run. If the current V is the second beat in a run, an interval statistic labelled VVC (V-V Couplet) is updated. If the beat is the third or greater V in the run, another interval statistic labelled VVR (V-V Run) is updated. [6] Finally, the sample number of the current beat is stored in order to determine the length of the next beat interval. The beat information is then written to the preCAL annotation file.

The PVC-Normal Module

The PVC-Normal Module of the CDP handles the V-N beat sequence. A flowchart of the functioning of this module is presented in Figure 4.6. Again assuming a subthreshold noise level and the appropriate beat sequence, the module begins by updating the N morphology statistics. Then a variety of operations are performed, depending upon the value of the run counter.

If the run counter has a value of zero, indicating that the PVC is isolated, then the module checks whether the PVC is interpolated. The time between the Normal beats flanking the PVC is computed and compared to the current state of the NN interval prediction. If the interval is within 15% of 1 NN interval, then the PVC is considered to be interpolated. The interpolated PVC counter is incremented and the Interpolated PVC/Total PVC ratio is updated [7]. If it is determined that the

[6] The rationale for this definition is explained in the final section of this chapter.

[7] This ratio, where Interpolated PVC refers to the total number of interpolated PVCs observed and Total PVC refers to the total number of PVC observed, is used to estimate the probability

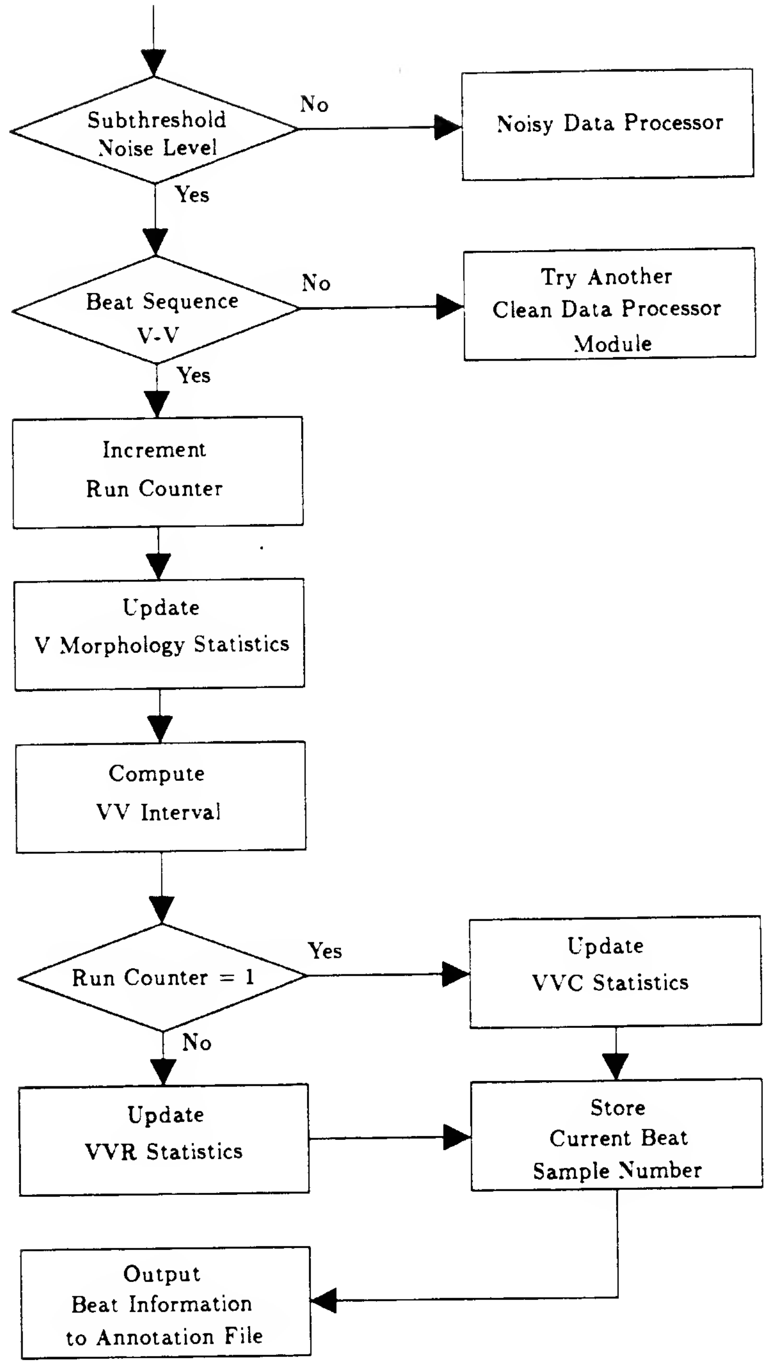


Figure 4.5: PVC-PVC Clean Data Processor Module Flowchart

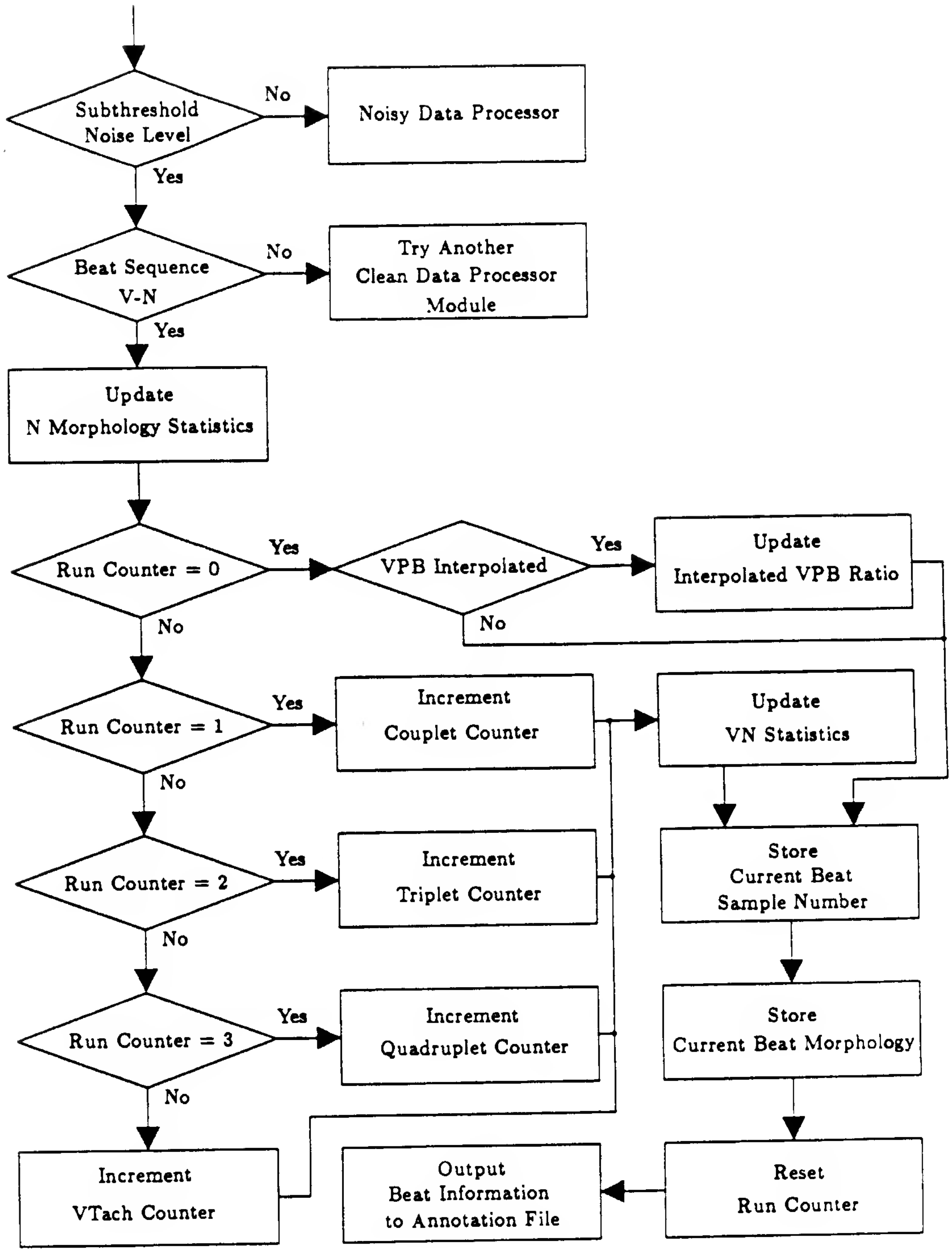


Figure 4.6: PVC-Normal Clean Data Processor Module Flowchart

PVC is not interpolated, then no special action is taken.

If the run counter has a value of one, then the Couplet Counter is incremented. If the run counter has a value of two, then the Triplet Counter is incremented. If the run counter has a value three, then the Quadruplet Counter is incremented. If the run counter has a value greater than three, then the VTach Counter is incremented.

Finally, the current beat sample number and classification (N) are stored and the Run Counter is reset. The beat information is then written to the preCAL annotation file.

4.4.2 Definition and Computation of the Knowledge Base

The knowledge base generated by preCAL consists of a series of averages, standard deviations, and counts used to characterize an ECG. During noise-free segments of ECG, the knowledge base is continuously updated based on the beat classifications assigned by ARISTOTLE. As mentioned earlier, the assumption is made by CALVIN that ARISTOTLE performs perfectly in the absence of noise and that the beat annotations are reliable. Once the noise level exceeds the threshold, the updating of the knowledge base ceases, and the current state is transferred from preCAL to CALVIN. The information is used by CALVIN to accurately analyze the noisy ECG segment. Following is a description of the items contained in the knowledge base.

The threshold used to determine the noise level at which CALVIN is activated is based on the average of the noise level estimates generated by ARISTOTLE during the 4 minute start-up period. ARISTOTLE quantifies the noise level on a scale of 0 (noise-free) to 16 (intense noise). PreCAL sets the threshold at 2 units above the computed average.

The QRS morphology feature used to characterize the ECG in this study is the output of a matched filter in the QRS detector of ARISTOTLE. The average

---

that a given PVC is interpolated.

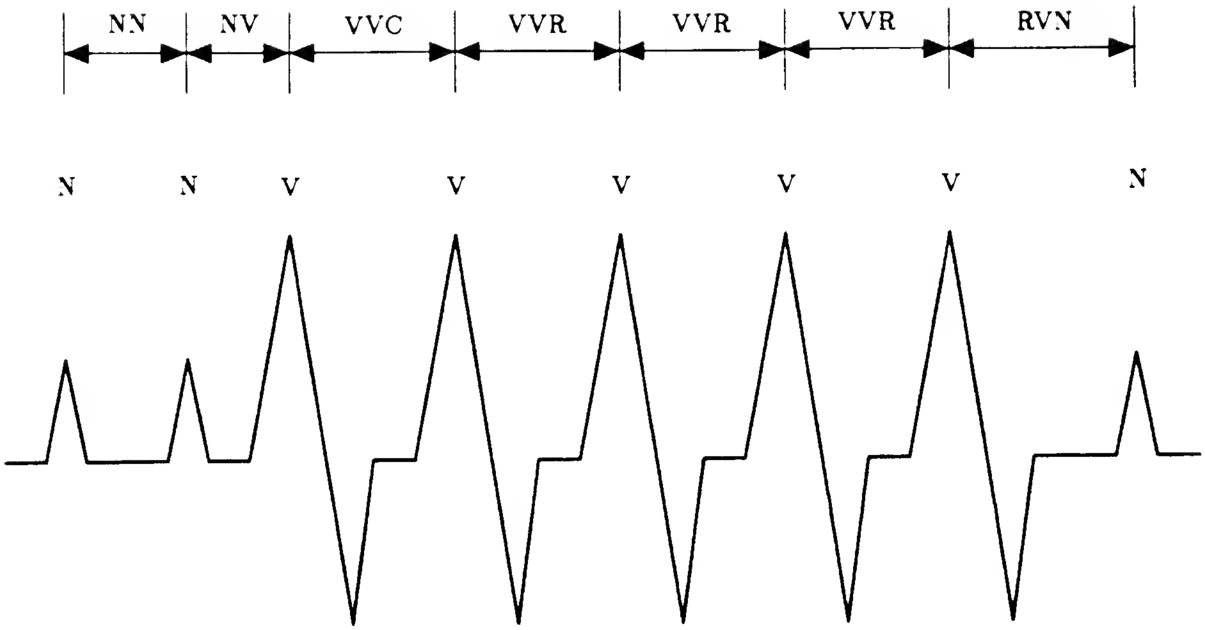


Figure 4.7: Definition of the VVC and the VVR Interval

and standard deviation of this feature is computed for all Normal beats and VPBs during the segments of ECG when CALVIN is in the Learn Mode.

The average and standard deviation is used to characterize the N-V interval. This approach was taken because we found that the N-V interval is not significantly rate dependent. The average and standard deviation is also used to characterize the first V-V interval in a run of VPBs (VVC), and subsequent V-V intervals (VVR). We chose to maintain separate VVC and VVR interval statistics, having observed that the first V-V interval is more variable than subsequent V-V intervals in a run of VPBs. The definition of these two intervals is illustrated in Figure 4.7.

The N-N interval prediction is determined using a first order low pass digital filter approach expressed by Equation 4.1, where RR_n and $\hat{RR}_n$ denote the actual and the predicted value of the N-N interval preceeding the n^{th} beat.

$$\hat{RR}_{n+1} = \hat{RR}_n + \alpha(RR_n - \hat{RR}_n) \qquad where\ \alpha\ =\ 0.3 \tag{4.3}$$

This approach to predicting the N-N interval has been shown to be superior to a moving average method [18]. The acceptable range used for the N-N interval is ±10-15% of the current prediction. This adaptive method of predicting the N-N interval responds rapidly to changes in heart rate and is therefore well suited for maintaining an accurate prediction.

The V-N interval in the case of an isolated VPB is often "compensatory". That is, the degree of prematurity of a VPB is matched by a corresponding delay in the subsequent Normal beat. This relationship is expressed by Equation 4.2, where VN_c is the compensatory V-N interval, NN is the local predicted interval between two Normal beats, and NV is the average forward VPB coupling interval. Obviously the N-N interval varies inversely with heart rate, while the N-V interval tends to remain fairly constant. Therefore VN_c is very dependent on heart rate. The statistics used to characterize the VN_c interval are shown in Equations 4.3 and 4.4. They were derived by applying basic probability principles for the mean and variance of sums.

$$VN_c = 2NN - NV \tag{4.4}$$

$$\begin{aligned} \mathbf{E}(VN_c) &= \mathbf{E}(2NN - NV) \\ &= 2\mathbf{E}(NN) - \mathbf{E}(NV) \\ &= 2\mu_{NN} - \mu_{NV} \end{aligned} \tag{4.5}$$

$$\begin{aligned} \mathbf{Var}(VN_c) &= \mathbf{Var}(2NN - NV) \\ &= 4\mathbf{Var}(NN) + \mathbf{Var}(NV) \\ &= 4\sigma_{NN}^2 + \sigma_{NV}^2 \end{aligned} \tag{4.6}$$

The NN and the NV intervals are treated as random variables with mean μ and variance σ^2. The N-N interval prediction is used for μ_{NN}, while 10% of the N-N prediction is used for σ_{NN} (the standard deviation of NN). The treatment of the NN interval as a random variable is reasonable if we assume that the heart rate is constant over an arbitrarily short time interval. The choice of 10% as the standard deviation is arbitrary.

The N-N and the VN_c interval predictions contained in Data Memory during noisy ECG segments are updated based upon the reclassified data stream generated by CALVIN. These rate dependent parameters must be continuously updated to correspond to the local heart rate. Otherwise, sudden changes in the heart rate would render them useless. The updating procedure is further explained in Chapter 6.

The statistics (average and standard deviation) for the V-N intervals subsequent to runs of PVCs are computed directly using the events observed during noise-free segments of ECG. PreCAL also maintains a count of all the occurances of couplets, triplets, quadruplets, and vtach (defined in preCAL as anything greater than four consecutive PVCs). The longest run of SVPBs is also recorded.

An internal count of the interpolated PVCs, the SVPBs, the total PVCs, and the total premature beats observed is maintained. The information transferred to CALVIN consists of two ratios. The first ratio is the percentage of PVCs that are interpolated (interp). This ratio estimates the probability that a given PVC is interpolated. The second ratio is the percentage of premature beats that are SVPBs (svpb). This ratio estimates the probability that a given premature beat is an SVPB.

4.5 Noisy Data Processor

The major function of the Noisy Data Processor (NDP) is to transfer the current state of the knowledge base to the preCAL annotation file at the beginning of and to determine the ending boundary of each noisy ECG segment. When three consecutive beats with noise estimates greater than the established threshold are observed, the NDP is invoked.

Several computations are then performed. First of all, the acceptable range of the current NN interval prediction is determined. The computed value is 3.3% of the NN prediction. The human expert protocol within Production Memory treats

this value as a standard deviation and the acceptable range is in most cases ± 3 standard deviations. Therefore the actual acceptable range is 10%. Secondly, the NV interval standard deviation is checked to assure that it is at least 3.3% of the average interval length. If the actual value is less than 3.3%, then 3.3% of the average NV interval is substituted as the NV standard deviation. Finally, the VN interval statistics are computed as described in section 4.4.2. The entire knowledge base is then transferred to the preCAL annotation file.

The classification of the beats within the noisy ECG segment is changed to Unknown (Q) by the NDP, since ARISTOTLE's beat classifications are unreliable in the presence of noise. The beat information is then transferred to the preCAL annotation file. The YAPS Production System will reclassify this noisy ECG data by applying the knowledge base and the human expert protocol. Once 20 consecutive beats with noise levels below the threshold are observed, control is transferred to the CDP.

Chapter 5

Implementation of the Human Expert Protocol with YAPS

This chapter will explain some of the basic principles of YAPS and the mechanics of how CALVIN was implemented with this production system. The next chapter will explain how the human expert protocol was extracted and implemented. It will develop an abstraction of the protocol to reveal how human experts analyze noisy ECG data.

5.1 Background

YAPS, Yet Another Production System , is an antecedent driven production system utilizing a discrimination (inference) net similar to the one used in OPS5 [19]. The conditional part of each rule (the Left Hand Side - LHS) is encoded within the inference net. When facts are added to the database, they are compared to the net, generating a list of rules whose LHS has been either partially or completely satisfied (bindings). The partial bindings are stored. The completely satisfied bindings constitute a conflict set. YAPS has a conflict resolution strategy, to be described in a later section, that it uses to decide which rule within the conflict set to invoke (or in Artificial Intelligence vernacular, which rule to fire). The action

part of the fired rule (the Right Hand Side - RHS) is implemented, modifying the database. The bindings are modified accordingly, generating a new conflict set. This iterative process is repeated until the conflict set is empty.

YAPS in general compares favorably to OPS5, providing greater flexibility in the patterns and tests allowable on the LHS and the actions allowable on the RHS. The YAPS Production System is implemented in Zeta LISP.

5.2 The AFTL Interface

As mentioned in chapter 4, PreCAL generates an annotation file as its output. The data format of the annotation file is incompatible with the YAPS production system. It must therefore be reparsed by the AFTL (Annotation File To List) Interface to a format compatible with YAPS.

5.2.1 Noisy Data Segment Extraction

YAPS is designed to handle lists of data with multiple fields of arbitrary length. The AFTL Interface converts the annotation file to a list of noisy segments of ECG. Within each of these segments is a list of potential ECG events, which are themselves lists with fields representing the data class, the beat classification, the sample number of the fiducial point, the noise level estimate, the matched filter amplitude, the pointer to the next beat, and the pointer to the previous beat. Most of the list items describing a beat are self explanatory. The data class field distinguishes the potential beats from the knowledge base entries and the other items in the YAPS database. The beat pointers will be explained in the next section. The reformatted data list structure is illustrated in Figure 5.1.

The interface software extracts the noisy ECG segments by first searching the preCAL annotation file for the NOTE annotations containing the knowledge base information. This indicates the beginning of the noisy data segment. In order for

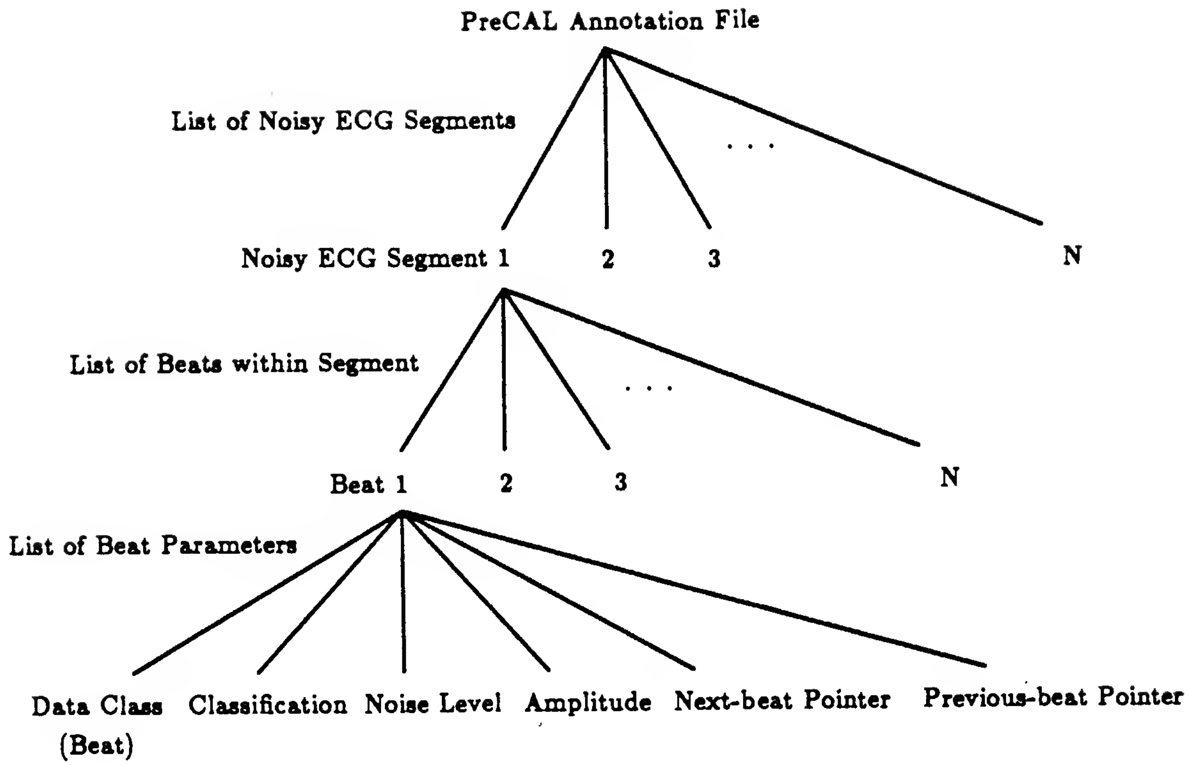


Figure 5.1: Reformatted Data List Structure

CALVIN to begin processing in the Assist Mode, a Normal beat must be identified at least 4 beats and not more than 8 beats prior to the first beat of the noisy segment. [1] All subsequents beats are reclassified to Unknown (Q).

The integrity of the first Normal beat of a noisy data segment is important, since CALVIN will base its initial decisions on the relative timing of this beat and the proximal events. Choosing a beat distant from the region of increased noise level raises the confidence in the accuracy of the Normal beat classification. A more sophisticated approach for a later version of CALVIN will be to have CALVIN assure the accuracy of the Normal classification by analyzing the beat context within the noise-free segment of data. If the beat classification if found to be questionable, the search for a Normal beat would proceed further into the noise-free region.

The end of the noisy ECG segment is determined by searching the the annotation file for 20 consecutive noise-free beats. If the 21^{st} beat is unclassified by ARISTOTLE (unknown - Q) the data segment is extended until a classified beat is observed. If the noise level should rise above the threshold during such an extension, an additional 20 consecutive noise-free beats is required to terminate the segment of data to be processed by CALVIN. This data extraction process is illustrated in Figure 5.2.

5.2.2 The Use of Beat Pointers

As will be illustrated in the next chapter, all of the rules implemented by CALVIN, except for the SITU mode rules that process all unknown events, search the database for patterns that contain both classified (NORM, VPB, APB) and unclassified (UNKNOWN) beats. The pattern matcher in YAPS does not take advantage of the sequential nature of the data. It attempts to match on every combination and

[1] If a Normal beat is not found within 8 beats from the beginning of the noisy data segment, CALVIN will commence processing in SITU Mode. This mode of processing will be explained in the next chapter. The inability to locate a Normal beat is caused by ARISTOTLE classifying a string of events as Unknown (Q) during a noisy ECG segment.

	N	N	N	N	V	N	N	Q	NOTE	Q	Q	Q	Q	Q	...	Q	Q	Q	N	N	N
Noise Level	2	1	3	1	3	1	1	4		3	6	4	7	8		0	1	0	2	1	1

	N	N	V	V	V	N	N	Q	NOTE	Q	Q	Q	Q	Q	...	Q	Q	Q	Q	Q	N
Noise Level	2	1	3	1	3	1	1	4		3	6	4	7	8		0	1	0	2	1	1

Figure 5.2: Noisy ECG Data Extraction Process

permutation of the data for every rule, a process that gives CALVIN a comatose appearance of inactivity. In order to restrict the domain of this search and eliminate unnecessary processing by CALVIN, each beat is assigned a pointer to the next and the previous beat. Regions of ECG data can be unambiguously defined by the use of beat pointers. This is illustrated in Figure 5.3. In Case A without the use of pointers, the 3 beat pattern defines 12 sequences from the database. Once pointers are introduced in Case B, the 3 beat pattern is satisfied by only one sequence from the database. This reduces the overhead processing by a factor of 12.

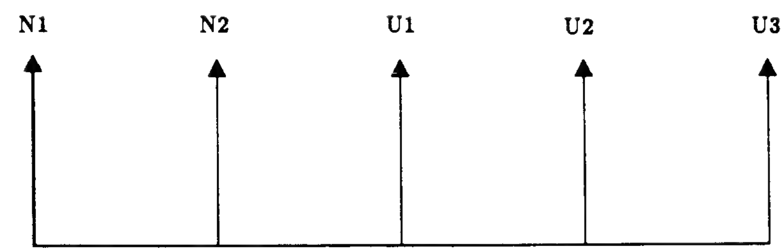
5.2.3 The Sliding Window Approach

Once the list of data is generated, it must be loaded into the YAPS database. Another technique that was found to increase the speed of CALVIN was to restrict the number of items in the database to be analyzed. The database window is comprised of an 8 event buffer of classified beats (CB Buffer) and an 8 event buffer of unclassified beats (UCB Buffer). [2] The Assist Mode initial processing state of CALVIN is 1 classified beat and 8 unknown beats in the window. As CALVIN begins to classify events, the CB Buffer is filled, while the UCB Buffer is maintained at 8 events by adding a new Unknown beat for every beat that is newly classified. Once the CB Buffer is filled to capacity, every newly classified beat replaces the oldest item in the buffer. This process is illustrated in Figure 5.4.

5.3 YAPS Conflict Resolution Strategy

In order to more fully understand the functioning and the modes of operation of CALVIN, one must understand the conflict resolution strategy utilized by YAPS. YAPS sorts the facts of all the bindings in the conflict set into two lists. One

[2] These are the default values. They can be altered by changing the value of the variables *slack-classified* (default 8) and *window-length* (default 8). This process is explained in the User's Guide.



A. No Beat Pointers
([Data Type] [Classification] [Sample Number])

(beat norm -id0)
(beat unknown -id1)
(beat unknown -id2)

Defines →

N1-U1-U2	N2-U1-U2
N1-U1-U3	N2-U1-U3
N1-U2-U1	N2-U2-U1
N1-U2-U3	N2-U2-U3
N1-U3-U1	N2-U3-U1
N1-U3-U2	N2-U3-U2

B. Next-beat Pointers
([Data Type] [Classification] [Sample Number] [Next-beat Pointer])

(beat norm -id0 -id1)
↓
(beat unknown -id1 -id2)
↓
(beat unknown -id2 -)

Defines →

N2-U1-U2

Figure 5.3: Definition of an Unambiguous Beat Pattern Using Pointers

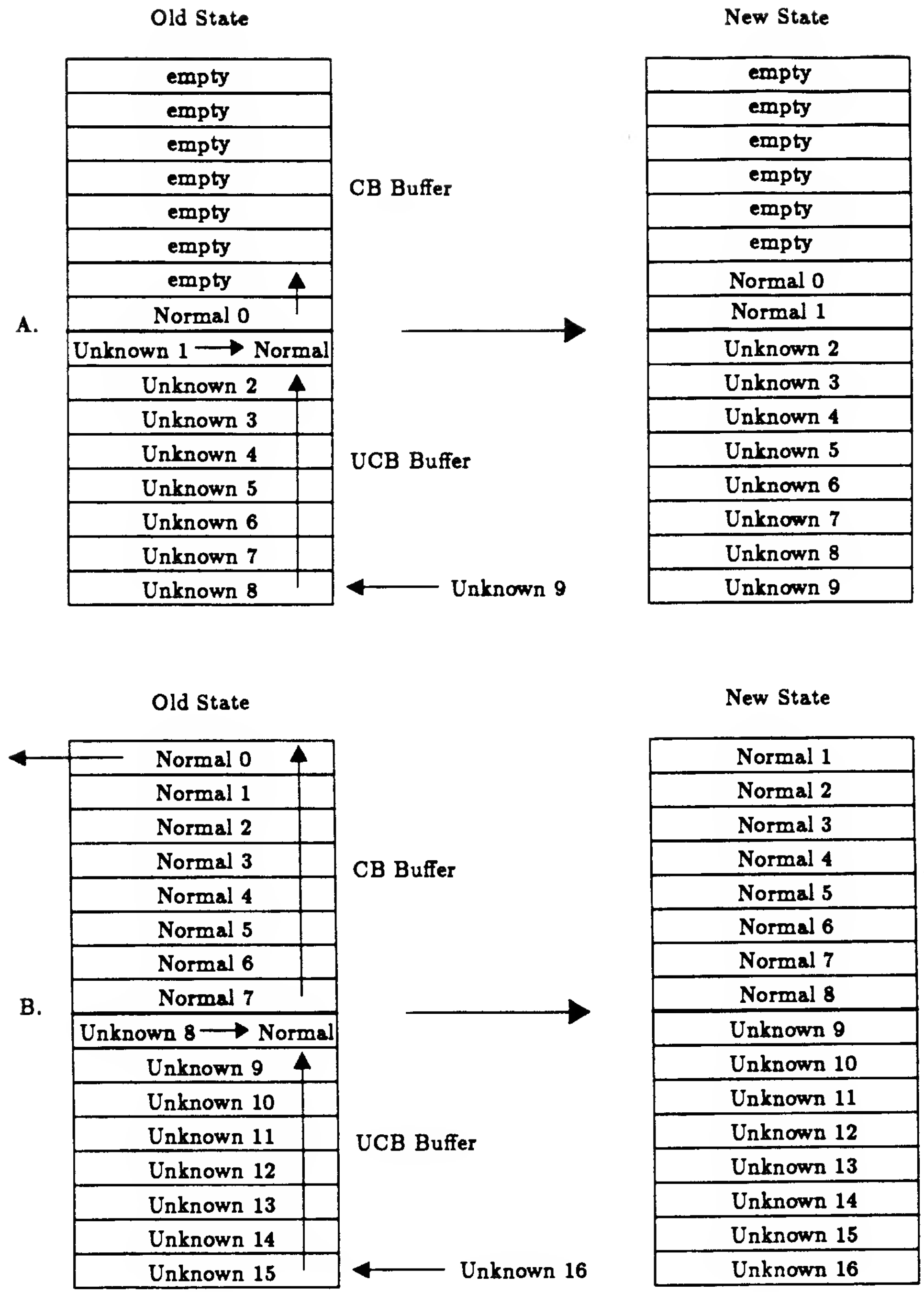


Figure 5.4: CALVIN Data Buffer Maintenance

list contains all the facts with the data class or keyword (the first field in a fact list) **goal**. The other list contains all the other facts used in the bindings. These lists of facts are sorted according to age, with the facts most recently added to the database placed first. The bindings are compared by looking at the facts in the **goal** list. The binding whose first fact was most recently added to the database is chosen by YAPS to be fired. If all of the first **goal** facts are of the same age (this implies it is the same fact), then successive facts in the list are compared for each binding. If a tie persists, then the binding with the longest list of facts is fired. If the lists are of the same length, then the second list of facts is compared in the same manner for each binding. If a tie persists for the second list of facts, then a random choice if made.

The YAPS Conflict Resolution Strategy is ideal for the implementation of the various modes of operation in CALVIN. It is also an ideal strategy for modelling the human experts' approach to analyzing noisy ECG's. These points will be explained further in the next chapter.

Chapter 6

The Human Expert Protocol

6.1 The Human Expert Approach

Putting aside for the moment the mechanics of how CALVIN operates, one vital question remains unanswered. How does the human expert approach analyzing noisy ECGs? The first step to answering this question involved actually observing the human experts performing this task.

As mentioned in Chapter 1, several interviews were held with the experts. They were asked to analyze both raw noisy ECG data and the beat annotation stream generated by ARISTOTLE. These sessions were recorded so that the rationale behind the decisions made could be reviewed. Once these sessions were completed, they were analyzed to extract a "universal" approach used by trained individuals in annotating noisy ECG recordings. The following observations deal mainly with the analysis of the beat annotation stream, except where noted. Review of the sessions revealed several common approaches used by the human experts while analyzing noisy ECGs. These included:

1. Events were classified based on their context and their timing relative to proximal events. Morphology information was used to a much lesser degree, especially while analyzing the beat annotation stream. The morphology infor-

mation contained within the raw ECG data was also of little use at the higher noise levels, except for VPBs which in many cases had larger amplitudes than the Normal beats.

2. Several hypotheses would be considered and the most probable one was selected.

3. The human experts were more assured of decisions based on a wider event context.

4. Only gradual changes in the heart rate were found acceptable. A regional R-R interval range (ie., local heart rate) was established and those beats that deviated from the acceptable range were classified as either VPBs, SVPBs, or glitches. The tape used in these sessions (AHA series 4001) represented a patient in NSR. Sessions were conducted with the annotation stream of patients in AF, but the experts were unable to analyze these in the absence of the beat morphologies (ie., the raw ECG data).

5. An acceptable range for the VPB coupling interval was established. This interval was assumed to be independent of the heart rate.

6. The clean ECG segment was checked for a compensatory pause subsequent to a VPB. If this was found to be the norm (as it is in most cases), then this timing pattern was used extensively during the analysis process.

7. The experts were reluctant to classify events as runs of VPB's (couplets, triplets, VTach) if no run was observed during the noise-free data segment.

8. Segments of data that were too difficult to annotate were skipped. A region would be found where accurate event classification could be resumed. The experts were usually able to then work backwards to classify portions of the skipped ECG data.

One very importat observation is that the human expert passes through several modes of analysis as the ECG noise level increases. At lower noise levels, the

morphology information is very reliable (or in the case of the raw ECG data, the expert is able to visually distinguish Normal beats from VPBs). Under these conditions, the human expert is able to make a majority of his decisions based on visual inspection[1]. As the noise level increases, the expert relies more heavily on the timing information (ie., begins to use his calipers) and relies less on the morphological information. At extreme noise levels, the expert losses confidence in his decisions and finds it necessary to skip segments of data.[2]

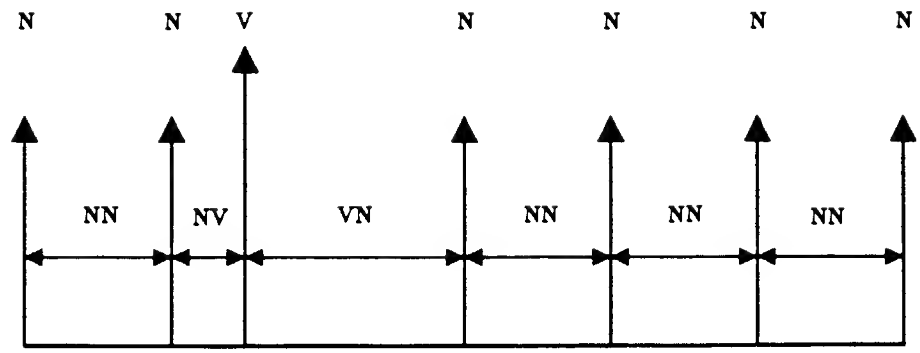
In order to further illustrate how the human expert analyzes the noisy ECG, let us examine several hypothetical situations where some of the above principles are applied. Consider the event pattern illustrated in Figure 6.1. The clean data segment is used to establish the knowledge base. Recall that the system preprocessor (preCAL) changes all of ARISTOTLE's beat annotations to unknown (U) during noisy ECG segments. The first unknown beat (U1) is 1 NN interval from N, while U2 is greater than 1 NN interval from N. U1 would be accepted as a Normal beat. A decision on event U2 would now be made in the context of U1 as a Normal beat. Note that the decision on U1 was based strictly on timing information.

Another straightforward example is presented in Figure 6.2. Event U1 is 1 NV interval from N, while event U2 is greater than 1 NN interval from N. The expert would therefore accept U1 as a VPB. A decision on event U2 would again be made relative to the previous decision on event U1 (VPB).

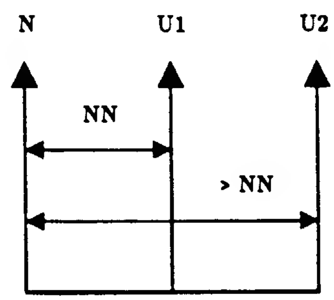
The two previous examples are very basic. Let us now examine a more complicated event sequence. Consider Figure 6.3. In this case, event U1 is 1 NV interval from N, while event U2 is 1 NN interval from N. Also, event U1 does not look like a VPB based on the morphology information available. Since the morphology

[1]Accurate analysis of ARISTOTLE's beat annotation stream by the human expert would require adequate separation of the Normal beat and the VPB matched filter amplitude distributions.

[2]Some of the human experts required constant encouragement to continue annotating the ECG at the higher noise levels for fear of making mistakes. Others proceeded with reckless abandon.



Clean Data Segment



Noisy Data Segment

True Beat Pattern N N U2

Figure 6.1: Timing Example

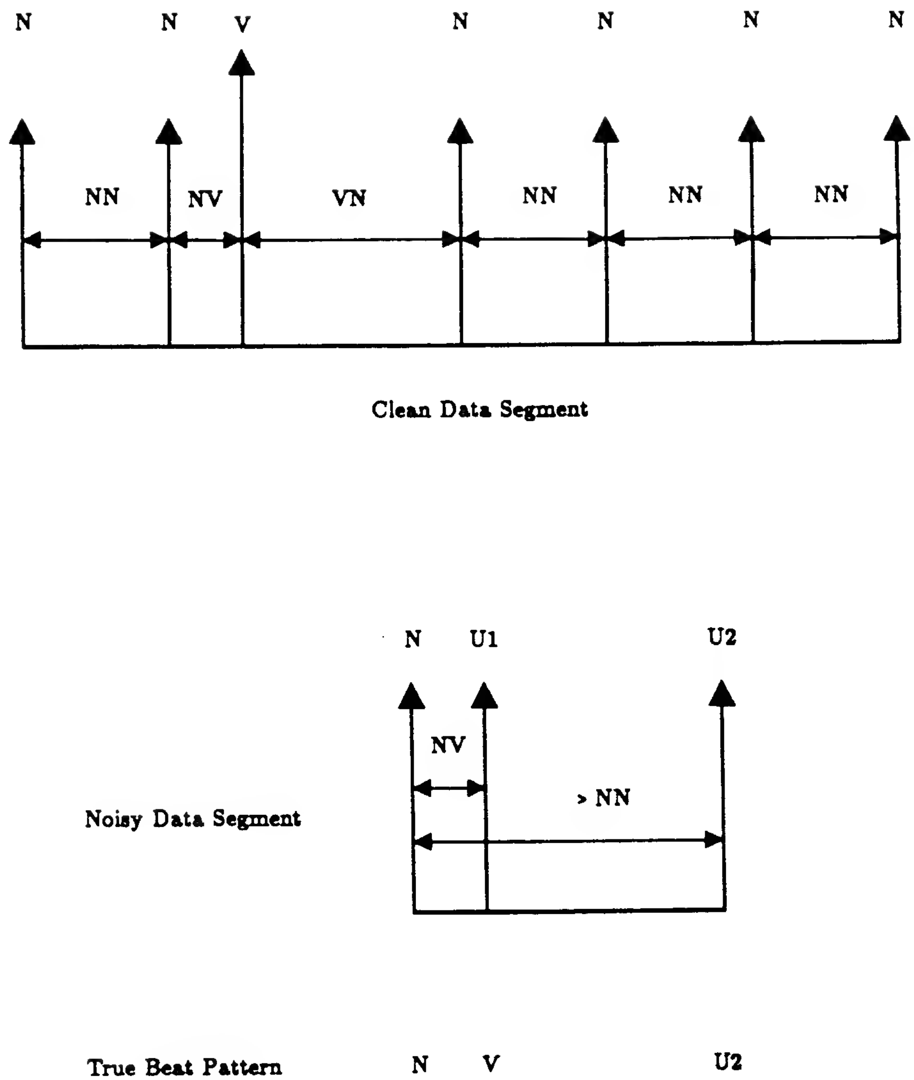


Figure 6.2: Timing Example

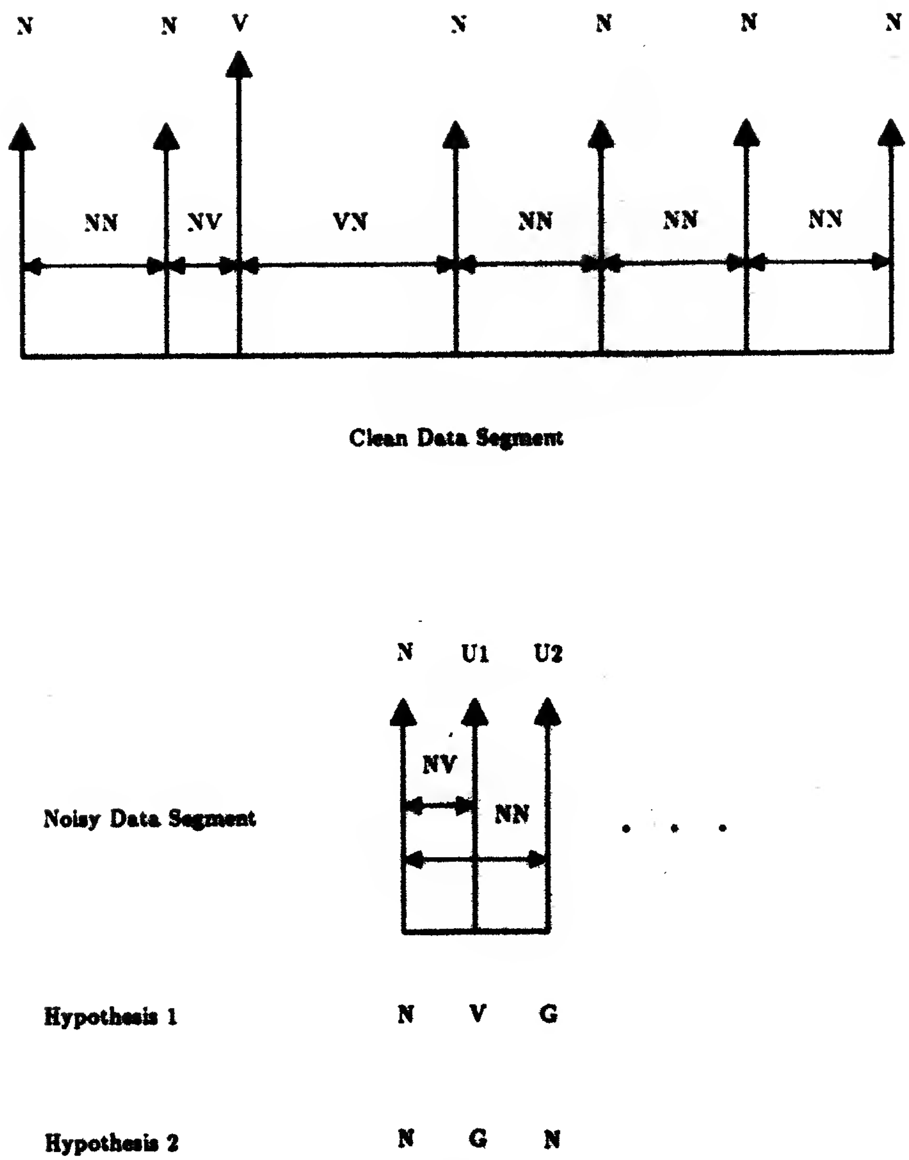


Figure 6.3: Timing Example

information is unreliable in the presence of noise [3], a confident decision ($p > 0.5$) cannot be made with this limited context. A wider event sequence is necessary for a definite decision to be made. This is provided in Figure 6.4.

Let us assume for the moment that we have not observed any SVPB's during the clean segments of ECG. Under Hypothesis 1 in Figure 6.4, event U1 is classified as a glitch since it is premature for a Normal beat and does not look like a VPB (which presently does not mean very much). Event U2 is 1 NN interval from N and is classified as a Normal. Event U3 is classified as an SVPB (the first to be observed) since it is premature for a Normal beat relative to U2, yet greater than 1 NV interval from U2. The fact that no previous SVPBs have been observed makes this hypothesis less likely to be true. Under Hypothesis 2, U1 is classified as a VPB in that it is 1 NV interval from N. U2 is less than 1 VN interval U1. Assuming that no previous couplets have been observed, U2 is classified as a glitch. Event U3 is 1 VN interval from U1 and is therefore classified as the compensatory Normal beat. Under these conditions, Hypothesis 2 would be accepted as the true beat sequence. A decision on event U4 would be deferred, requiring a wider event context.

There are situations were a definitive event classification cannot be made. In Figure 6.5, event U1 is 1 NV interval from N. Event U2 is 1 NN interval from N. Event U3 is both 1 VN interval from U1 and 1 NN interval from U2. U1 does not look like a VPB. Because of the unreliability of the morphology information and the ambiguity of the timing pattern, a confident decision cannot be made between Hypothesis 1 and Hypothesis 2.

The ECG rhythm prior to the unknown events under consideration may contain useful information for the handling of an ambiguous event sequence. If, for example, the prior beat context was an episode of bigeminy, then this would influence the current decision. If presented with the ambiguous event sequence of Figure 6.5

[3] It should be noted that with the matched filter output of ARISTOTLE,

$$p(Event = VPB|Morphology = VPB) > p(Event \neq VPB|Morphology \neq VPB)$$

This fact is strictly empirical.

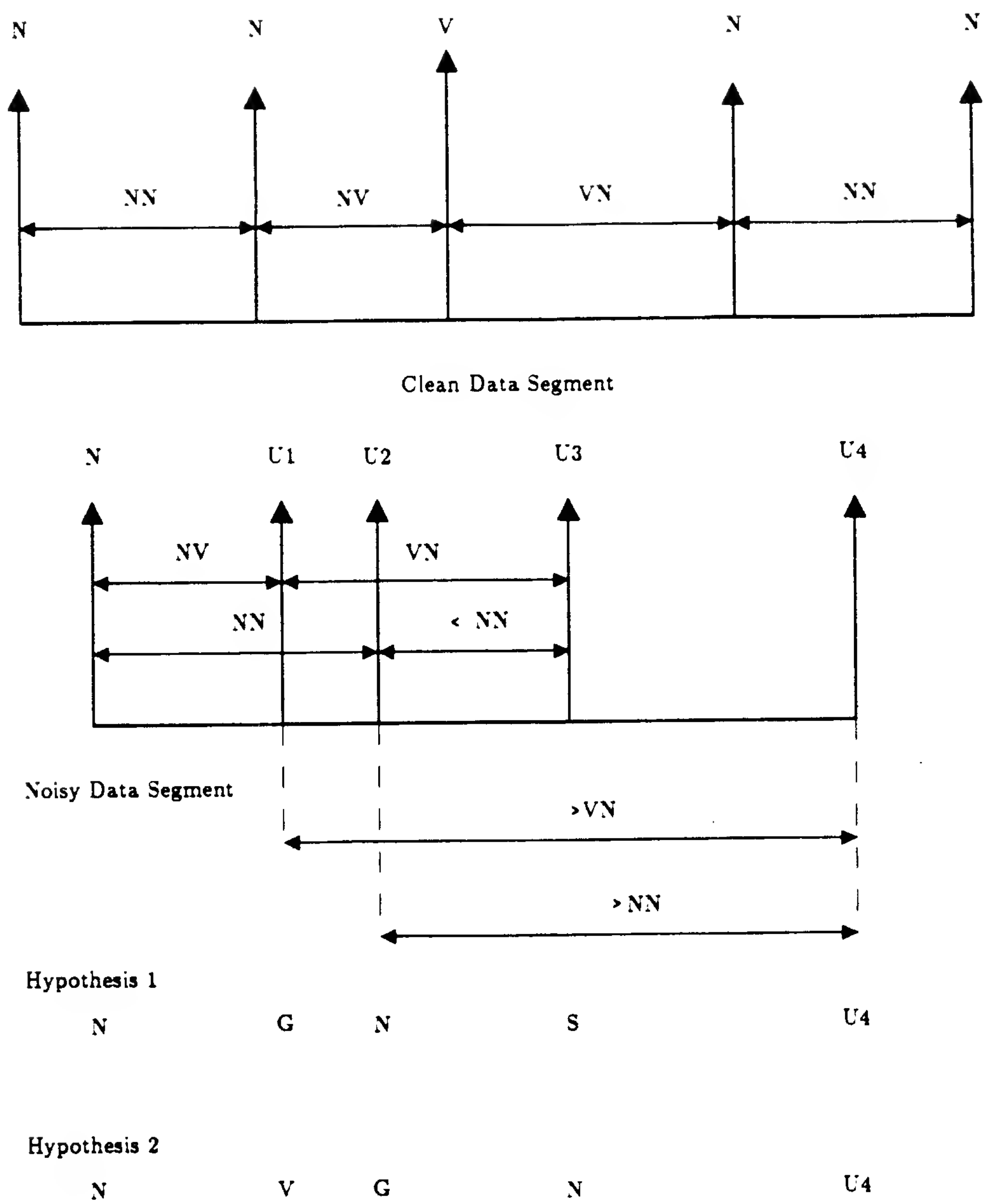


Figure 6.4: Timing Example

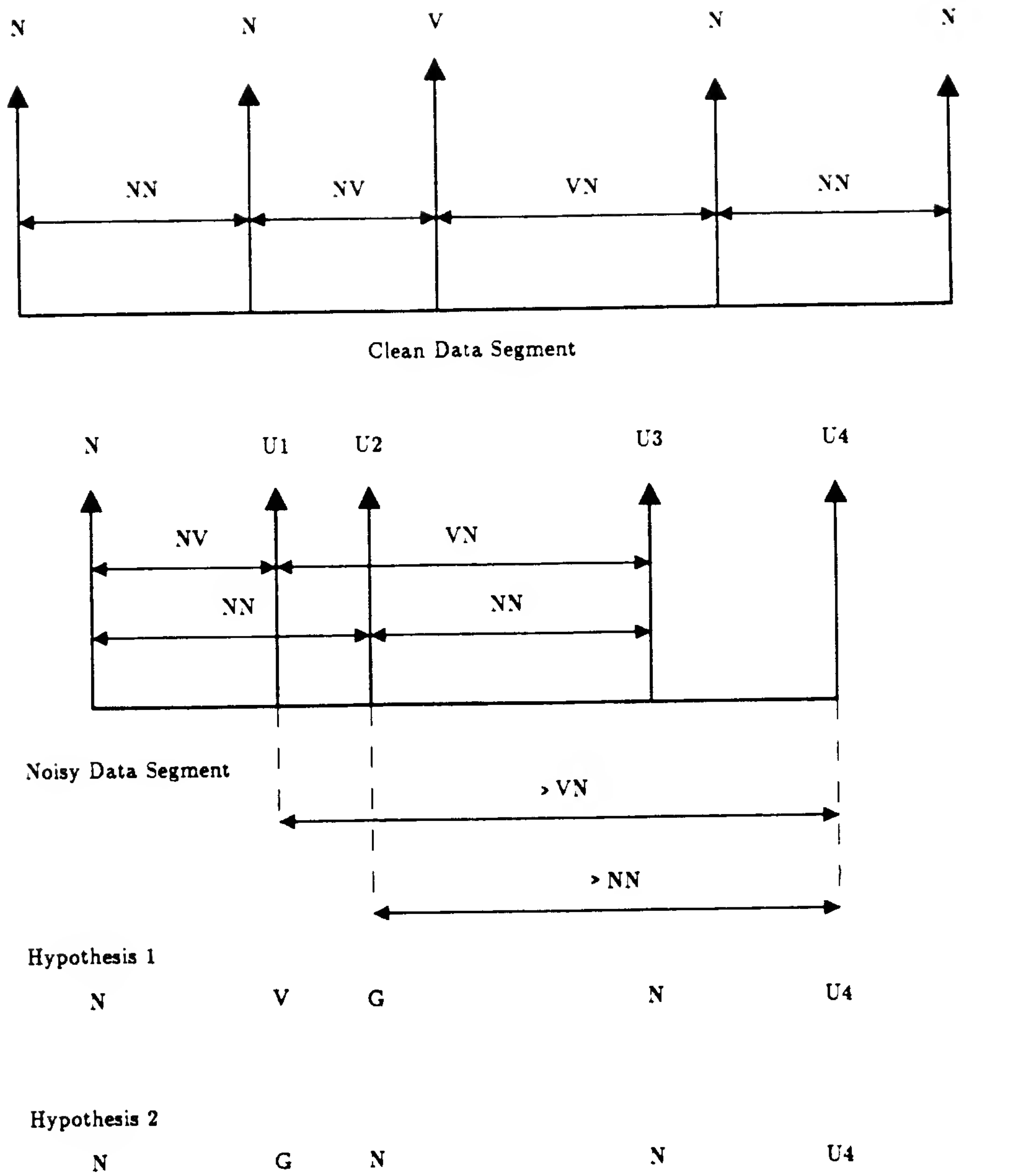


Figure 6.5: Timing Example

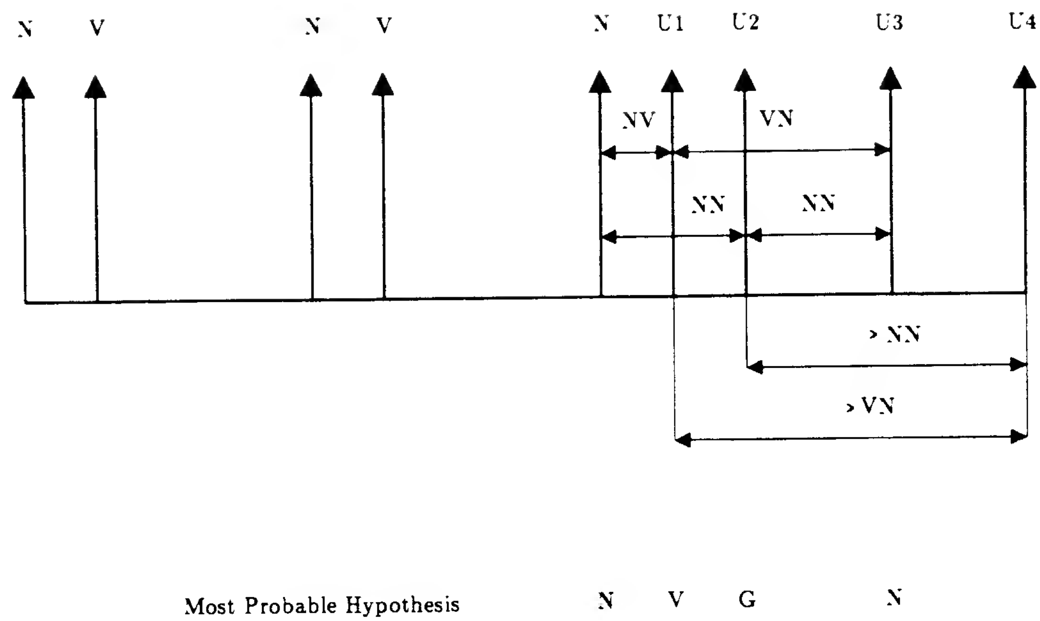
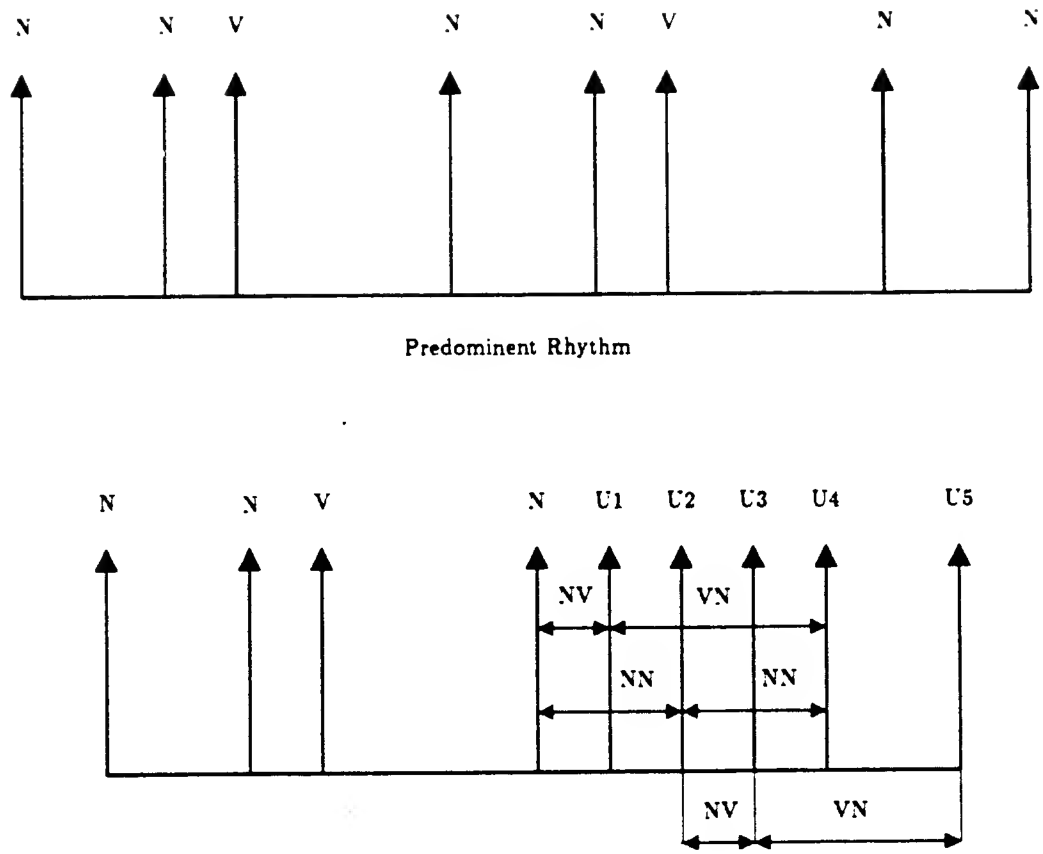


Figure 6.6: Timing Example

in the context of bigeminy, the expert would accept Hypothesis 1 as the most likely beat sequence. This is illustrated in Figure 6.6. In the context of trigeminy, the most likely beat sequence would depend on whether the Normal beat at the classification front (ie., the last classified beat) was the first or the second Normal beat in the N-N-V sequence. This is shown in Figure 6.7. The tendency of the human expert is that if a particular rhythm has been identified, decisions are biased towards maintaining that rhythm.

Once the human expert skips a segment of data due to an excessive noise level, the classification problem becomes more complex. The decisions are no longer based on the event timing relative to a set of classified beats. The expert searches the event data stream for a pattern that matches a template constructed from previously obtained timing information. This is illustrated in Figure 6.8. In this



Hypothesis 1	N	V	G	G	N	U
Hypothesis 2	N	G	N	G	N	U
Hypothesis 3	N	G	N	V	G	U

Figure 6.7: Timing Example

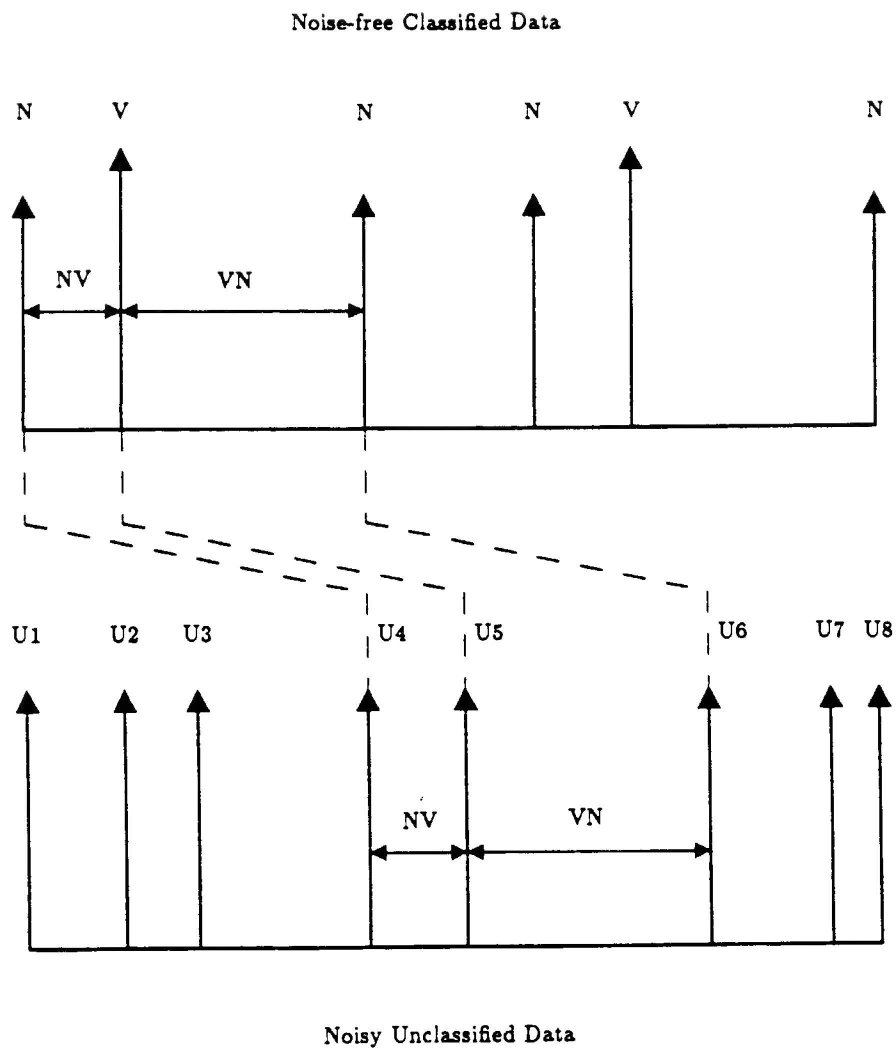


Figure 6.8: N-V-N Template Matching

case, the expert observes that VPB's are followed by a compensatory pause. He therefore constructs an N-V-N template and searches the unknown beats for an event sequence that matches this template. One could also construct and utilize an N-N-N template as illustrated in Figure 6.9. Some of the experts actually drew marks on a card representing the beat timing and moved the card through the annotation data stream in search of a match. Once a match was found, the experts would resume annotation at the newly classified beats. When noisy segments were skipped in the raw ECG data, the experts usually resumed annotation at a VPB that was easily distinguishable from the surrounding noise due to their large amplitude and width.

Since we are dealing with noisy ECG segments, the template matching procedure had to be able to tolerate intervening noise glitches. In order for a match to be declared, certain criteria must be satisfied regarding the timing of the noise glitches In Figure 6.10, the potential N-V-N pattern is interrupted by a noise glitch during the V-N segment. The timing of the glitch is such that it is greater than 1 NN interval from event U1 (potential Normal beat) and greater than 1 NV interval from event U2 (potential VPB). If we assume that no couplets have been observed during the noise-free ECG, then we would conclude that U3 is a glitch and that U1, U2, and U4 represent the N-V-N beat sequence.

Consider the similar scenerio with the N-N-N template, shown in Figure 6.11. In this case, the glitch (U3) is greater than 1 NN interval from U1 and less than 1 NV interval from U2. One would also have to rule out the possibility that U3 is a compensatory Normal beat relative to event U1. In this situation, one would have to utilize the morphology information to assure that U1 was not a VPB, although this would not be very reliable. If the distance between U3 and U1 is greater than any previously observed VN interval, then this fact would rule out U3 as a compensatory Normal beat.

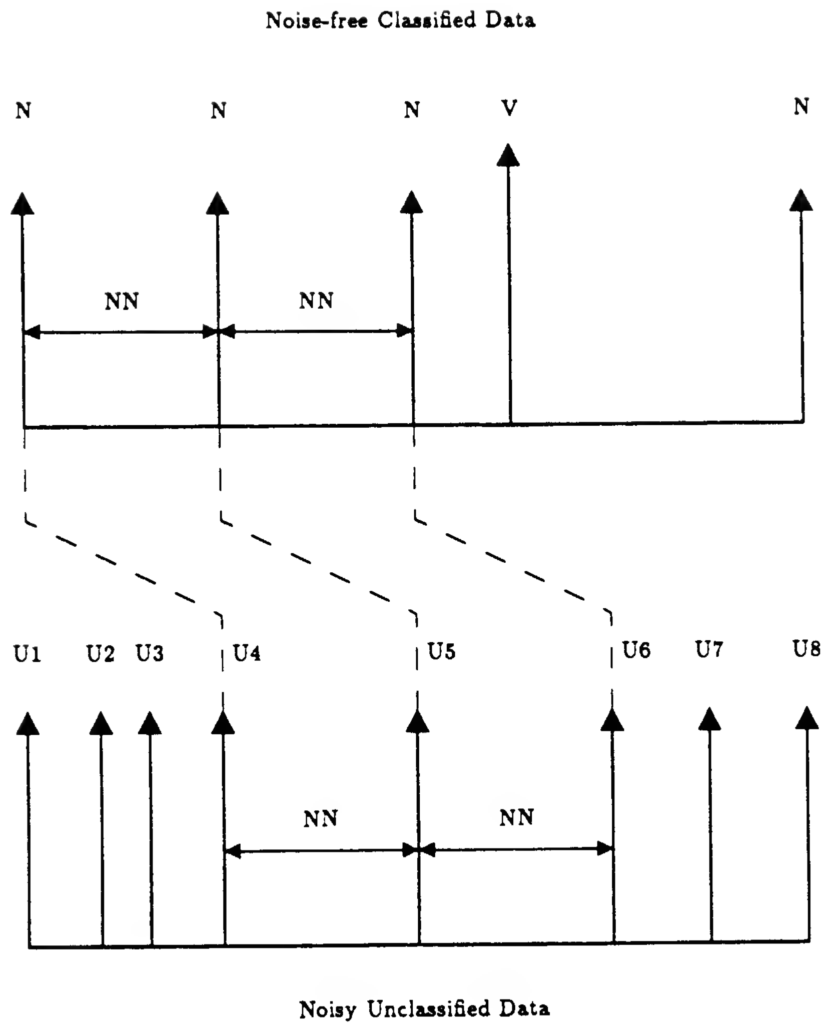


Figure 6.9: N-N-N Template Matching

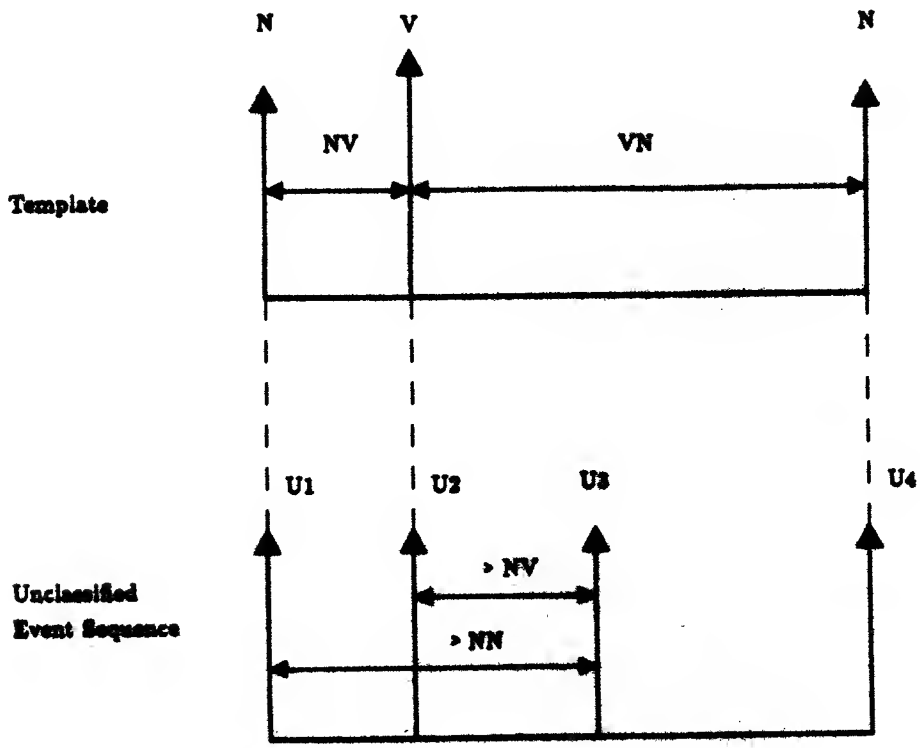


Figure 6.10: N-V-N Template Matching with Noise

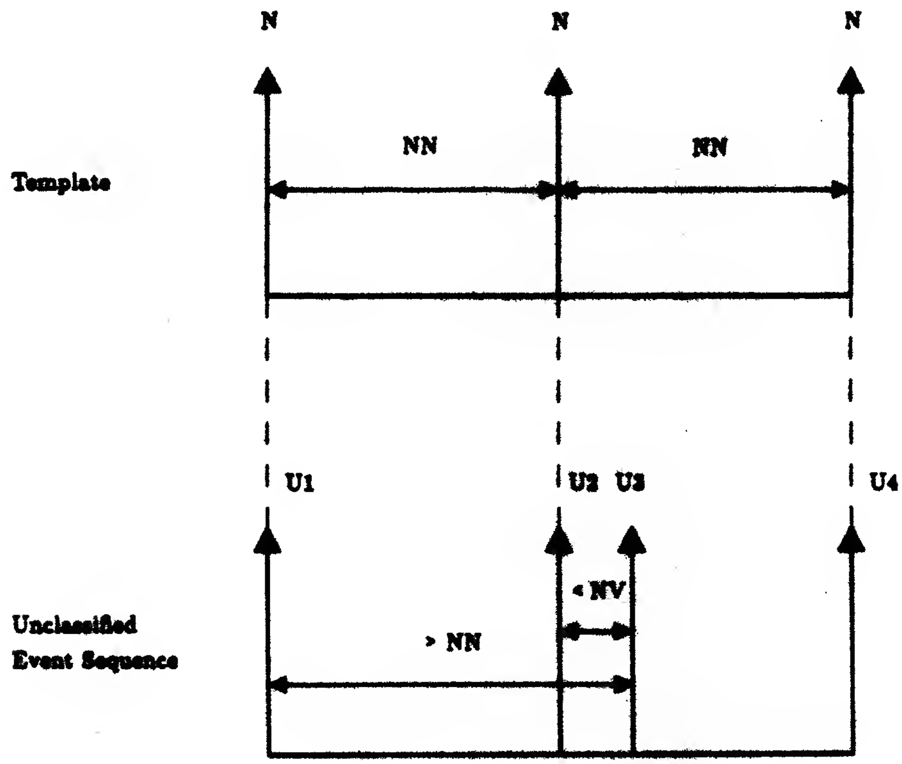


Figure 6.11: N-N-N Template Matching with Noise

6.2 CALVIN Rule Structure

Once the basics of the human expert approach were identified, a small subset of the rules were generated based on hypothetical event sequences. CALVIN was then run on a noisy ECG tape (AHA series 4001). When CALVIN was unable to annotate a given segment of the ECG, the event sequence was identified and a rule was generated to handle it, provided the sequence was unambiguous. This process was continued until a "high" level of system performance was attained.

The rules consist of four major components, the beat pattern, the knowledge base subset, the analytical section and the action section. A representative example of a coded rule is shown in Figure 6.12. Section A of this rule represents the beat pattern being analyzed, which in this case is a Normal beat followed by three unknown beats.[4] The fields for a beat represent the data class, the beat classification, the sample number, the noise level, the beat amplitude, the pointer to the next beat, and the pointer to the previous beat. Notice that the pointers have all of the fields of the top level beat. The pointers are used to unambiguously define a sequential set of events. Each beat in the pattern, except for the last beat, specifies the sample number of the next beat in the pattern. The dashes with no variable name after them represent "don't care" fields. In the case of rule 14a, we are not interested in the previous beat field (the last field) of the top level beats. The field must be held by a dash in order for the YAPS pattern matcher to search the database for a fact with 7 fields, regardless of how many fields the user is interested in.

Section B of this rule represents the components of the knowledge base applied by the rule. In this case, the statistics for the NN interval, the NV interval, the VN interval, and the VPB amplitude are used. This rule also used the fact that SVPBs represent zero percent of the total premature beats observed (svpb 0.0). The list "(goal calvin assist aristotle)" is used to establish the Assist Modes in CALVIN. The establishment of the various modes in CALVIN will be explained in a later

[4] An event is a beat until proven otherwise.

```
(p calvin_14a

A.      (beat norm -id0 -nlev0 -amp0 (beat - -id1 - - - -) -)
        (beat unknown -id1 -nlev1 -amp1 (beat - -id2 - - - -) -)
        (beat unknown -id2 -nlev2 -amp2 (beat - -id3 - - - -) -)
        (beat unknown -id3 -nlev3 -amp3 - -)

B.      (goal calvin assist aristotle)
        (nn -nnavg -nnsd -nnsd3 -nnsd5)
        (nv -nvavg -nvsd -nvsd3 -nvsd5)
        (vn -vnavg -vnsd -vnsd3 -vnsd5)
        (vamp -vampavg -vampsd -vampsd3 -vampsd5)
        (svpb 0.0)

test    (< (// (abs (- (- -id1 -id0) -nvavg)) -nvsd) 3)
        (< (// (abs (- (- -id2 -id0) -nnavg)) -nnsd) 3)
C.      (< (// (abs (- (- -id3 -id2) -nvavg)) -nvsd) 3)
        (> (// (abs (- -amp1 -vampavg)) -vampsd) 3)
        (< (// (abs (- -amp3 -vampavg)) -vampsd) 3)

=>      (fact glitch -id1 -nlev1 -amp1)
        (remove-beat 2)
D.      (modify 3 beat norm)
        (int-update 6 7 8 (- -id2 -id0))
        (new-facts 2)
        )
```

Figure 6.12: Example of a Coded Rule – Calvin_14a

section.

The analysis is performed in Section C. Intervals between specified events are computed and compared to the averages compiled by the preprocessor. In most cases a window of ± 3 standard deviations about the mean of a given parameter is considered to be the acceptable range of the value. A window of ± 5 standard deviations is used to establish whether a computed value is out of range for consideration as a given parameter. There are however exceptions to these rules. The VN interval distribution, computed from the NN and the NV interval statistics as demonstrated in section 4.3.2, tends to be very wide. Therefore ± 2 standard deviations about the mean is used as the cutoff between the acceptable and the unacceptable range of values. In some cases, depending upon the event context, the acceptable range of values for the NV interval are from -5 standard deviations to +3 standard deviations about the mean value. Due to the usually wide distribution of the amplitude statistics, ± 3 standard deviations is used as the cutoff between the acceptable and the unacceptable values.

In Rule 14a (Figure 6.12), the first test in the analytical section checks whether the interval between the normal beat and the first unknown beat is within 3 standard deviations of the mean NV interval. The next test checks whether the interval between the normal beat and the second unknown beat is within 3 standard deviations of the mean NN interval. The third test of event timing checks whether the interval between the third unknown beat and the second unknown beat is within 3 standard deviations of the mean NV interval. The two final tests check the amplitudes of the first and the third unknown beat. The first test assures that the amplitude of the first unknown beat is out of the acceptable range for VPB's (greater than 3 standard deviations from the mean). The second test assures that the amplitude of the third unknown beat is within 3 standard deviations of the mean VPB amplitude.

Sections A, B, and C represent the predicate of this rule or the LHS. Section D represents the action part or the RHS of this rule. There are several new functions

that have been added to the YAPS production system for this specific application. The first of these is the **modify** routine. This function changes the classification of a specified beat. The first argument specifies the location, or the line within the rule, of the beat to be modified. The second and the third argument specify the data class (beat) and the new beat classification respectively. In Rule 14a (Figure 6.12), the **modify** function is applied to the second unknown beat on line 3. The beat classification is changed from unknown to norm (Normal). The function also refreshes the modified beat and the flanking beats in order to enable the fired rule for data in the region under analysis. [5]

The **remove-beat** function is used to remove events that have been determined to be glitches from within the data structure. The only argument that it requires is the line number of the event to be removed. The procedure removes the event from the data structure by redirecting the "next-beat" pointer of the left flanking beat to the right flanking beat. The "previous-beat" pointer of the right flanking beat is redirected to the left flanking beat. This process is illustrated in Figure 6.13. Notice that prior to removing the first unknown beat in Rule 14a, a fact "glitch" is created with all of the attributes of the removed beat except for the pointers. This information allows the location of the detected ECG artifact to be noted within CALVIN's annotation file.

The NN and the VN interval statistics, maintained in Data Memory during the Assist Mode, are continually updated by CALVIN. Since the VN interval is dependent upon the heart rate, changes in the NN interval statistics must also be reflected in the VN interval statistics. This is accomplished by CALVIN with the int-update function. The first three arguments of the int-update function are the line numbers within the rule of the NN, the NV, and the VN interval statistics respectively. Since it has been determined in Rule 14a that the first unknown beat is a glitch and the second unknown beat is a Normal, the NN interval statistics are updated based on the length of the interval between the second unknown beat (id2)

[5] The necessity of the refresh procedure is explained in the YAPS User's Manual in the section describing the conflict resolution strategy.

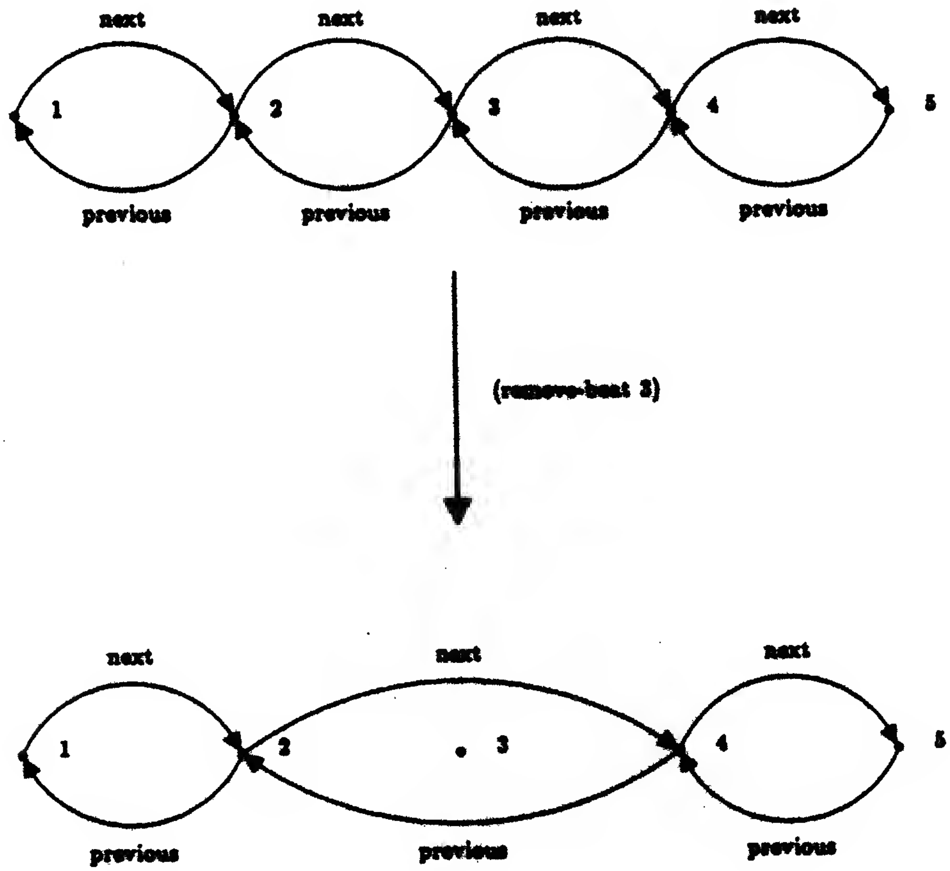


Figure 6.13: Remove-Beat Procedure

and the Normal beat (id0). The length of this interval (- -id2 -id0) is the fourth argument to **int-update**. The statistics are updated as described in Equations 4.1-4.4 of section 4.3.2. Note that the NN interval statistics must be updated first in order for the change to be reflected in the VN interval statistics. The line number of the NV interval statistics is supplied since they are required to compute the VN interval statistics. The NV interval statistics are not updated by **int-update**.

New unknown events are added to the data buffer in proportion to the number of beats classified by CALVIN. This is accomplished by the **new-facts** function. The routine is called with one argument specifying the number of sequential unknown events to add to the UCB Buffer. In Rule 14a, two unknown beats were classified, therefore **new-facts** is called with an argument of 2.

The last function added to the YAPS Production System is **clean-up-facts**. A review of CALVIN's modes of operation will help illustrate the use of this function. CALVIN operates in three basic modes, Learn Mode, Assist Mode, and SITU Mode as illustrated in Figure 6.14. During Learn Mode, the knowledge base is compiled. When the noise level exceeds the threshold, CALVIN enters Assist Mode and begins classifying the events in ARISTOTLE's annotation stream. When CALVIN is unable to process a segment of data, it enters SITU Mode. During SITU Mode, CALVIN skips the difficult segment of ECG and searches the unknown events for a recognizable timing pattern. CALVIN first searches the 8 events in the UCB Buffer. If a recognizable pattern is not found, a new unknown beat is added to the database and the search is repeated. This process is continued until CALVIN finds a pattern it recognizes. Then CALVIN reenters Assist Mode and beat classification is resumed.

For each new unknown beat added to the database, the oldest beat (classified or unclassified) is removed from the database. This is to maintain the total window length (CB Buffer and UCB Buffer) at 16 beats. This is accomplished with the **new-facts** function with a second non-NIL argument. If one were to call **new-facts** under these circumstances without the second argument, the UCB Buffer

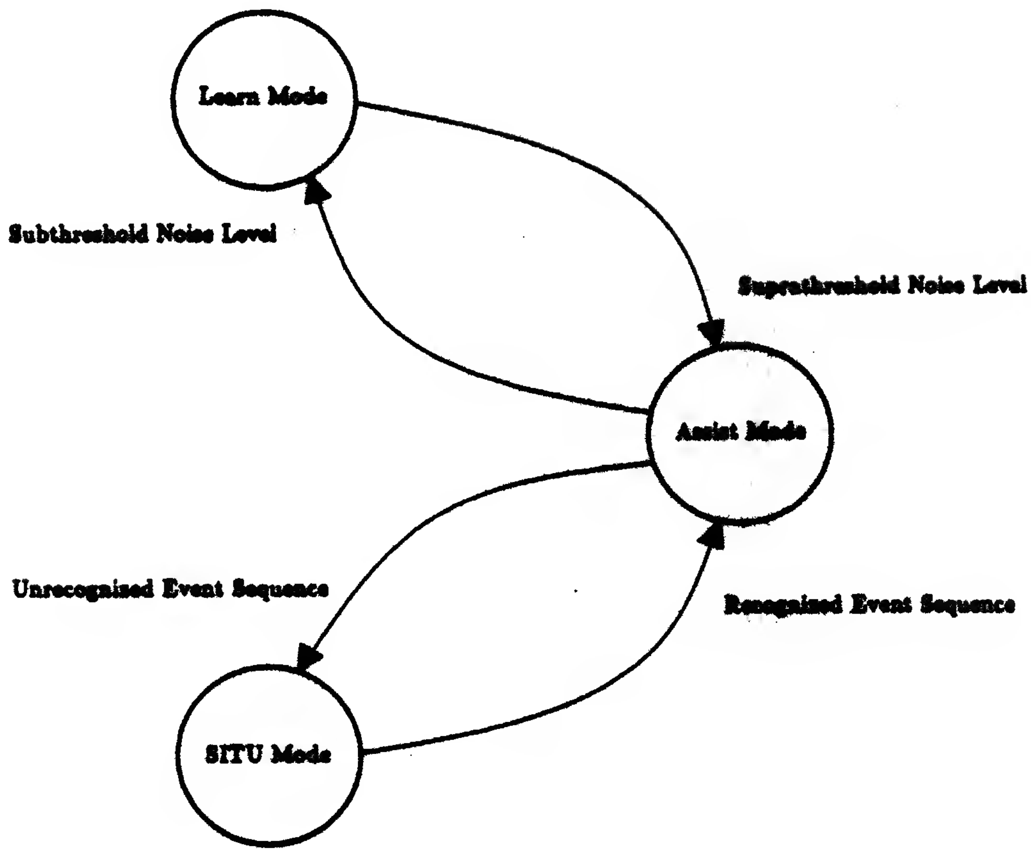


Figure 6.14: CALVIN Mode Transition Diagram

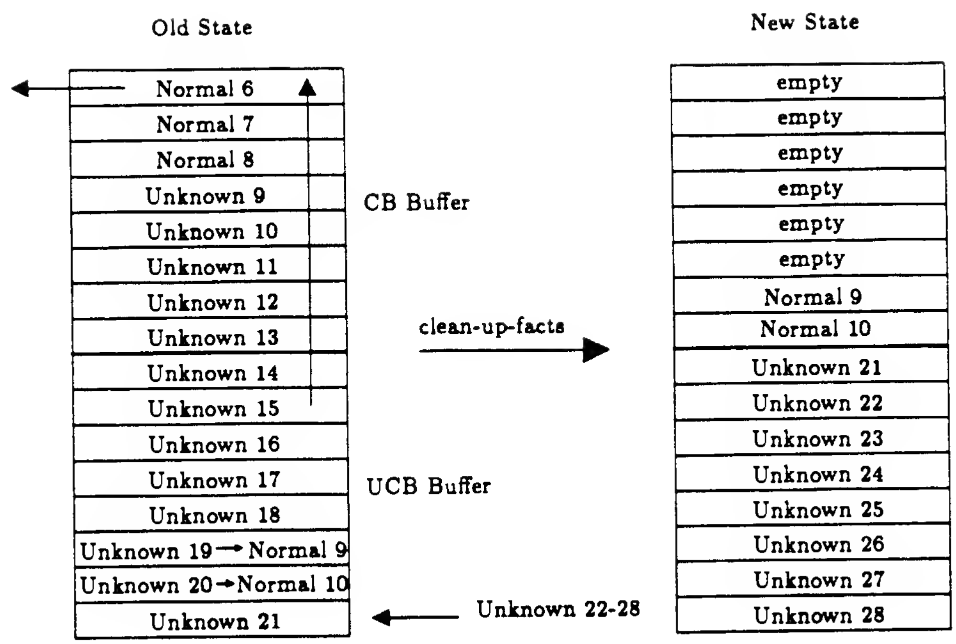


Figure 6.15: Action of **Clean-up-facts**

---

would increase in length indefinitely.

The result of this search process in SITU Mode is a discontinutiy of the regions of ECG classified by CALVIN. Since CALVIN is not capable of working backwards, the segment of events up to the second set of beats classified in SITU Mode must be removed from the database. This is accomplished with the function **clean-up-facts**. **Clean-up-facts** is called with no arguments. Its action is illustrated in Figure 6.15. The beats classified by the SITU Mode rule become the sole occupants of the CB Buffer. All of the preceding beats are discarded from the database. Since the SITU mode rules classify all but the last event in its beat pattern, the UCB Buffer will contain one unclassified event. Therefore **new-facts** must be called

1. (goal calvin assist aristotle)
2. (calvin change mode situ)
3. (calvin advance window situ)

Figure 6.16: Facts Added to Database During Calvin Boot Sequence

---

with an argument of 7 in order to fill the buffer.

6.3 Establishment of the Assist and the SITU Modes

The rules for the Assist Mode and the SITU Mode reside in the same production memory. The rules for the SITU Mode must be disabled during the Assist Mode and enabled when the Assist Mode conflict set is empty, implying that CALVIN has halted processing due to the inability to process the current ECG segment. This is accomplished by taking advantage of the conflict resolution strategy utilized by YAPS.

When CALVIN is initially booted, the three facts presented in Figure 6.16 are added to the database in the order shown. Each of CALVIN's Assist Mode rules has as the first item in the knowledge base subset the fact "(goal calvin assist aristotle)". As mentioned in Chapter 5, YAPS gives priority to rules that match facts with the keyword "goal". None of the SITU mode rules match facts with this keyword. During the Assist Mode, only one SITU Mode rule is capable of firing. This rule, **calvin_changemode_situ** illustrated in Figure 6.17, matches only on fact "(calvin change mode situ)". It is therefore of lower priority than all of the Assist Mode rules. This rule is always triggered during Assist Mode, but not fired until the Assist Mode conflict set is empty. All of the other SITU mode rules are disabled, since they require the fact "(situ mode indicator)", which is created on

```
(p calvin_changemode_situ

        (calvin change mode situ)

⇒       (remove 1)
        (fact situ mode indicator)
        )
```

Figure 6.17: SITU Mode Rule **Calvin_changemode_situ**

the RHS of the rule **calvin_changemode_situ.**

Once SITU Mode is invoked by the creation of the fact "(situ mode indicator)", the Active Mode rules are implicitly disabled requiring a new set of classified beats in the data buffer for triggering. The rule **calvin_changemode_situ** disables itself by removing the fact "(calvin change mode situ)" from the database. The priority structure of the SITU Mode productions is such that all of the rules dedicated to classifying events take priority over the rule **calvin_advance_situ** (Figure 6.18), which is responsible for adding new unknown beats to the database. This situation is just the opposite to that in the Active Mode. In this case, the rule of the lowest priority (calvin_advance_situ) continually fires and reenables itself until one of the higher priority SITU Mode rules is triggered. The dedicated SITU Mode rule classifies the recognized beat sequence, disables the SITU Mode by removing the fact "(situ mode indicator)", and reenables the rule **calvin_changemode_situ** by creating the fact "(calvin change mode situ)". The Active Mode is invoked, since new classified beats are added to the database by the SITU Mode rule. An example of a SITU Mode rule dedicated to classifying events is shown in Figure 6.19.

```
(p calvin_advance_situ

        (calvin advance window situ)
        (situ mode indicator)

-->     (refresh 1)
        (new-facts 1 T)
        )
```

Figure 6.18: SITU Mode Rule Calvin_advance_situ

```
(p calvin_3_situ

        (beat unknown -id0 - -amp0 (beat - -id1 - - - -) -)
        (beat unknown -id1 - - (beat - -id2 - - - -) -)
        (beat unknown -id2 - - - -)

        (situ mode indicator)
        (nn -nnavg -nnsd -nnsd3 -nnsd5)
        (nv -nvavg -nvsd -nvsd3 -nvsd5)
        (vn - - - -)
        (vamp -vampavg -vampsd -vampsd3 -vampsd5)

test    (< (// (abs (- (- -id1 -id0) -nnavg)) -nnsd) 3)
        (< (// (abs (- (- -id2 -id1) -nnavg)) -nnsd) 3)
        (> (- -id1 -id0) (+ -nvavg -nvsd3))
        (> (- -id2 -id1) (+ -nvavg -nvsd3))
        (> (// (abs (- -amp0 -vampavg)) -vampsd) 3)

⇒       (fact calvin change mode situ)
        (modify 1 beat norm)
        (modify 2 beat norm)
        (int-update 5 6 7 (- -id1 -id0))
        (remove 4)
        (clean-up-facts)
        (new-facts 7)
        )
```

Figure 6.19: Dedicated SITU Mode Rule

Chapter 7

The Evaluation of CALVIN

7.1 Selection of the AHA Database Tapes

Eight ECG tapes were selectively chosen from the AHA database to evaluate CALVIN, based on the current capabilities of the system. The tapes represented patients in normal sinus rhythm with isolated, unifocal VPBs. Care was taken to select tapes on which ARISTOTLE made a minimal number of errors in the absence of noise.

CALVIN is limited in its potential ability to process more complex rhythm classes by the morphology descriptor (matched filter output). The matched filter output does not provide enough information to make the differentiation between Normal beats and VPBs in the presence of noise. Another problem with this descriptor is that it is not very consistent during high levels of noise.

It has been emphasized that the timing information is most important in analyzing noisy ECGs, yet when one is dealing with more complex rhythms, beat morphology takes on a more important role. Consider the event sequence illustrated in figure 7.1. In this case, we have a VPB followed by 3 unknown beats. Assume for the moment that we have observed several episodes of triplets in previous segments of noise-free ECG. Based on the timing information, U1 and U2 could

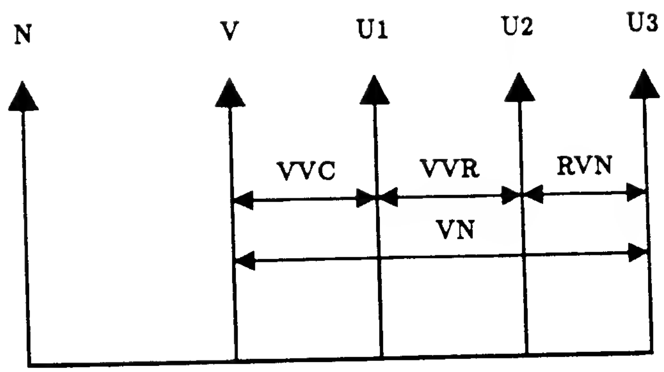


Figure 7.1: Event Sequence Requiring Morphological Information to Resolve

---

be VPBs or V could be an isolated VPB with U3 representing the compensatory Normal beat. In the absence of reliable morphology information, a sound decision on the true event sequence cannot be made. CALVIN will currently classify any beat that is shorter than 1 VN interval from a VPB as a Glitch.

Another issue is that of SVPBs. There are some instances where the timing information is sufficient to distinguish an SVPB from a VPB. But consider the situation where the coupling interval distributions for SVPBs and VPBs have significant overlap. The morphology information now becomes quite important in the classification of premature beats.

Because of the limitiations imposed by the current morphology descriptor, rules have not been developed to handle many complex beat patterns. The AHA Database Tapes used to evaluate CALVIN were therefore chosen in accordance with this early stage of the rule development. The purpose of the evaluation is not to exhibit CALVIN's current domain of application, which is very limited. The following evaluation results represent an attempt to reveal the potential usefulness of this approach.

Another important factor in the choice of the AHA Tapes was that the timing characteristics had to be such that PVCs could be distinguished from Normal beats based on timing alone. Otherwise, the timing information that CALVIN relies on heavily in the classification of beats during noisy ECGs would be useless. Rapid heart rates produce NN intervals that do not differ significantly from the NV intervals. Tapes were therefore chosen with heart rates in the range of approximately 60 to 80 beats per minute, providing adequate separation between the NN and the NV interval distributions.

7.2 Evaluation Protocol

The evaluation protocol used to evaluate CALVIN is illustrated in Figure 7.2. Electrode motion noise was added to the ECGs [10] beginning at 5 minutes into the tapes, at a level that was shown to cause significant degradation in the performance of ARISTOTLE operating alone. An example of a noisy ECG signal is shown in Figure 7.3. The noise was added in 2 minute bursts separated by 2 minute noise-free regions. The test was conducted over the first 20 minutes of each database tape (including the initial 5 minute learn period). Two of the tapes were a part of the development set for the system (tapes 4001 and 4009).

The corrupted ECG data was then processed by ARISTOTLE, which generated an annotation file containing the beat classifications. This annotation file was then reprocessed by CALVIN, which generated another annotation file with modified beat classifications. Both of these annotation files were then compared to the truth annotation file for each database tape, resulting in 2 sets of performance statistics.

7.3 Results

The performance measures used were the QRS and PVC sensitivity and positive predictivity. The statistics were compiled only during those segments of ECG that

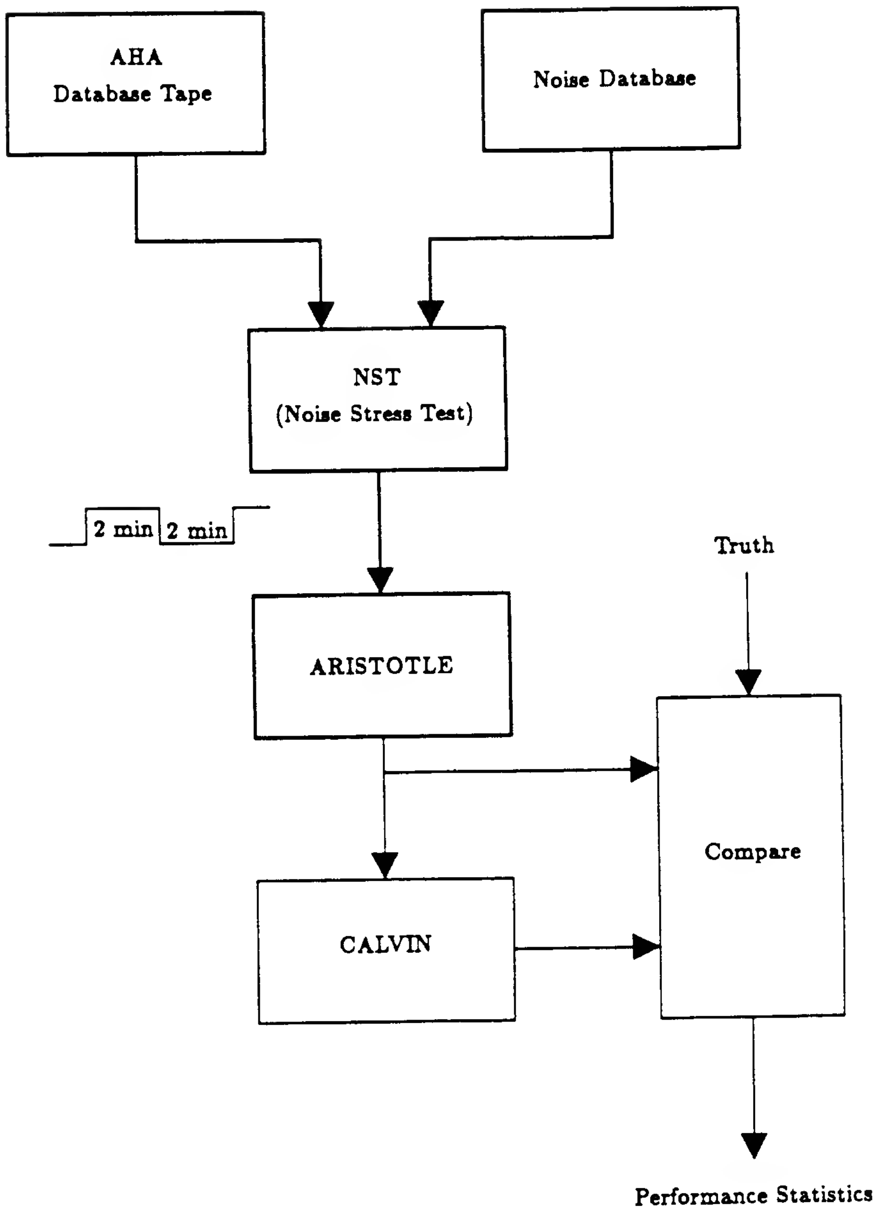


Figure 7.2: Evaluation Protocol used to Evaluate CALVIN

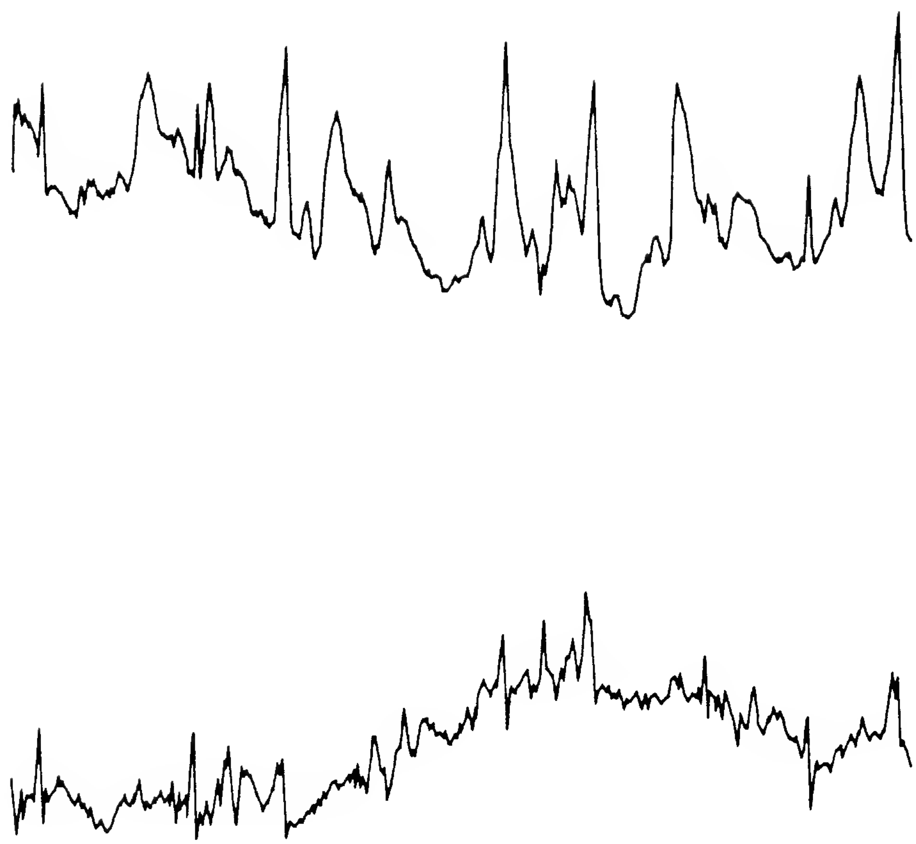


Figure 7.3: Quality of ECG Used to Generate Results.

The figure shows a 5 second segment from AHA Tape 4009 with added electrode motion noise. The noise was added at an equivalent level to the other ECG tapes used in this study.

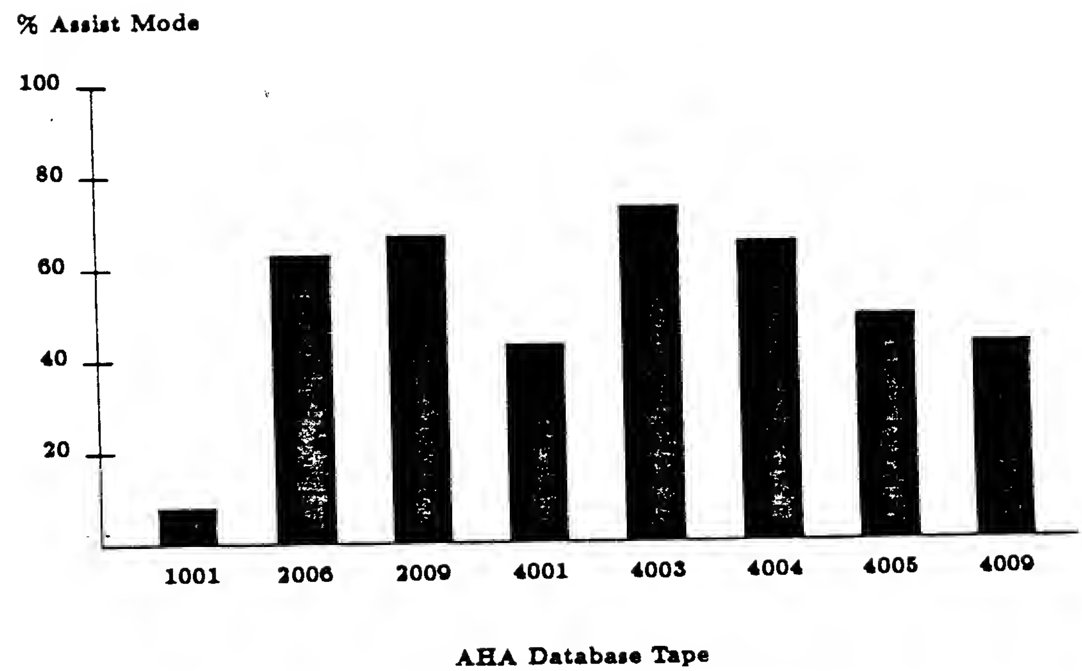


Figure 7.4: Percentage of Processing Time in Assist Mode

CALVIN processed in the Assist Mode. CALVIN processed an average of 58.0% of the noisy ECG segments in Assist Mode, while 42.0% of the data segments were too noisy to process (SITU Mode). The results for the individual AHA Tapes are shown in Figure 7.4. The only noteworthy result is for Tape 1001, which did not have VPBs. The rules were developed under the implicit assumption that VPBs are observed during the noise-free data segments. CALVIN will have to be modified to handle the situation where no previous VPBs are observed. [1]

The QRS performance statistics are presented in Figure 7.5. In almost every case, CALVIN slightly lowered the overall QRS sensitivity. Since CALVIN was actively discarding events determined to be false positive detections, it is evident that some true beats were discarded. The average QRS sensitivity for ARISTO-

[1] It is felt that the necessary modifications will be relatively minor.

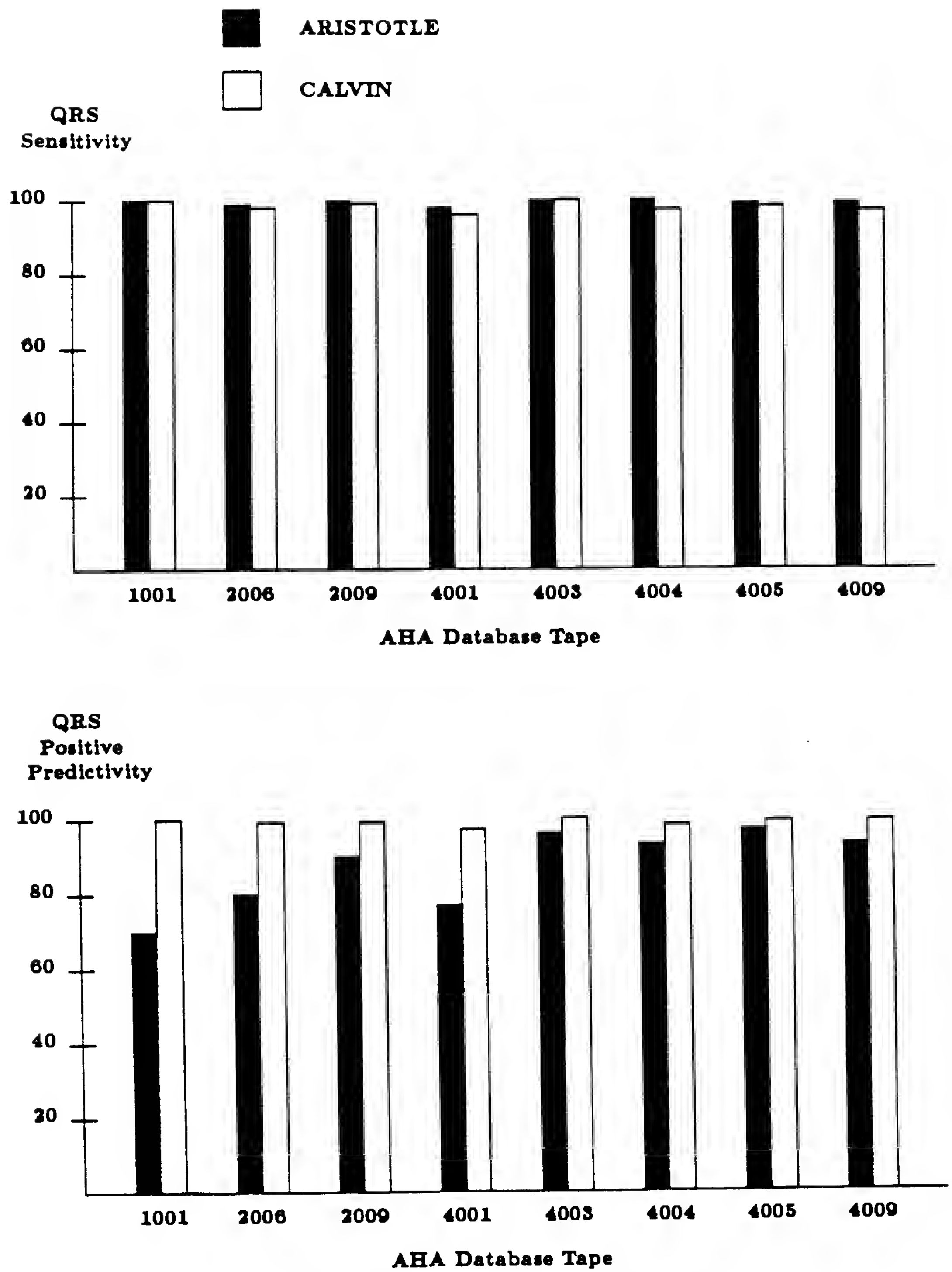


Figure 7.5: QRS Performance Statistics.

ARISTOTLE's performance is represented by the black bars, while CALVIN's performance is represented by the open bars.

TLE operating alone was 99.0%, while that for CALVIN (i.e., ARISTOTLE plus CALVIN) was 98.0%.

CALVIN improved the QRS positive predictivity for every ECG tape used, illustrating the effectiveness of CALVIN in removing the false positive detections of ARISTOTLE. The average QRS positive predictivity for ARISTOTLE operating alone was 87.0%, while that for CALVIN was 99.0%.

The most striking results were in the PVC statistics, presented in Figure 7.6. CALVIN significantly outperformed ARISTOTLE in terms of both the sensitivity and the positive predictivity. Note that for tape 1001 the PVC sensitivity is undefined and the PVC positive predictivity is zero, since there were no VPBs on this tape. ARISTOTLE had 22 false positive VPB detections on this tape, while CALVIN had only 1. For tape 2006, ARISTOTLE and CALVIN performed at the same level in terms of the VPB sensitivity (97.4%), yet CALVIN far outperformed ARISTOTLE in terms of the VPB positive predictivity (97.4% versus 30.6%). Tapes 4005 and 4009 revealed comparable performance for the two algorithms in terms of the VPB positive predictivity, while CALVIN far outperformed ARISTOTLE in terms of the VPB sensitivity for these two tapes. The average PVC sensitivity for ARISTOTLE operating alone was 60.0% while that for CALVIN was 95.0%. The average PVC positive predictivity for ARISTOTLE operating alone was 58.2% while that for CALVIN was 88.7%. These results show that CALVIN is able to effectively correct both false positive and false negative VPB detections.

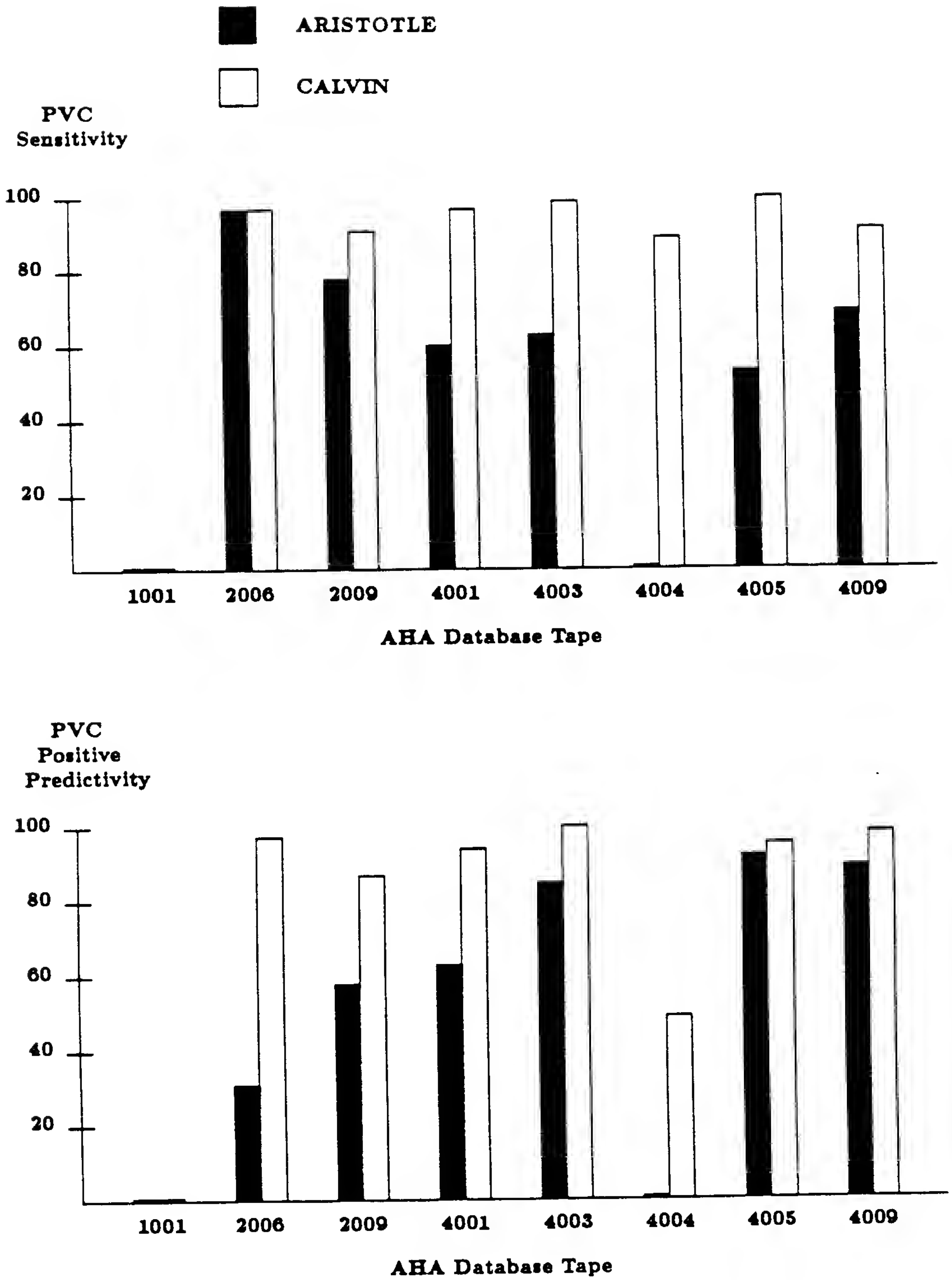


Figure 7.6: PVC Performance Statistics.

ARISTOTLE's performance is represented by the black bars, while CALVIN's performance is represented by the open bars.

Chapter 8

Discussion

8.1 Ongoing Development

CALVIN is still in the early stages of development. There are several developmental issues that need to be addressed. First of all, the current version of CALVIN is not able to operate in real time. CALVIN requires 1 to 2 hours (depending upon the noise level) to process 2 minutes of noisy ECG data. [1] The slow processing speed is attributable to the YAPS production system not taking advantage of the sequential nature of the data and therefore performing unnecessary overhead processing. The continued development of CALVIN requires a faster production system, since as the number of rules increase, the speed of the system decreases. Further customization of YAPS is necessary to decrease the processing time of CALVIN. Also, the development of a new production system designed specifically for this application is now under consideration.

One might also consider implementing the human expert protocol in "C" during the final stages of development when major changes to the rules would not be anticipated. This system approach would conceivably amount to representing the rules as a series of if-else if/then statements placed sequentially from the highest to

[1] This does not include the processing time of the Walking Interface, which can vary depending upon such things as weather conditions, physical health status, and tape drive availability.

the lowest priority rule. This would provide both real time processing and complete compatibility with the preprocessor (preCAL), also written in "C".

Currently, CALVIN is not capable of handling rhythms such as triplets, quadruplets, VT, and supraventricular tachycardia (SVT). It can handle SVPBs and couplets to a very limited extent. In order to handle such rhythms, a more advanced morphology descriptor is needed. We have emphasized that event timing is most important in analyzing noisy ECGs, yet when one is dealing with complex rhythms like VT or SVT, the only way to distinguish them from a string of false positive QRS detections is to rely more heavily on event morphology. The matched filter output alone does not provide enough information in most cases to correctly identify such rhythms under noisy conditions. Once a more sophisticated morphology descriptor is integrated into the system, further sessions will be conducted with the human experts and rules will be developed to handle these important rhythm classes.

One potential extension of the CALVIN Project is the identification of recurrent beat patterns and their subsequent use as templates while analyzing noisy ECG data. The Human Experts used this approach to a moderate degree (especially with the N-V-N beat sequence) by making timing marks on a card and sliding it through the data stream to assist in the identification of specific beat patterns. CALVIN constructs its own templates from the timing information within the knowledge base, but it does not record the occurence of specific, recurring patterns. Once a recurrent pattern is identified, the timing information could be autocorrelated with that of the unknown event sequence to provide additional information for more accurate beat classification by CALVIN. This approach should prove to be very powerful in analyzing noisy ECG data.

8.2 Power of the Approach

There are several reasons one would expect CALVIN to outperform a conventional arrhythmia detector in analyzing noisy ECGs. Firstly, CALVIN's decision making process is adaptive. The bias of the system is dependent upon both the local rhythm information and the data contained in the knowledge base. For example, the system is quite reluctant to classify an event as a couplet if no couplets have been observed previously and a "better" hypothesis exists that is more consistent with the information in the knowledge base.

Secondly, conventional arrhythmia detectors tend to be relatively limited in the number of beats analyzed at any given time. CALVIN analyzes the data using a 16 event window (8 classified beats and 8 unknown events). This allows for a much greater degree of contextual analysis of the noisy ECG data. For example, an ambiguous event sequence can be resolved in some cases by observing that it exists in the context of bigeminy. The process of current decisions being based on previous ones makes for a more self-consistent decision making process.

Finally, CALVIN represents a sound model of the dynamic thought process used by the Human Expert while analyzing noisy ECG data. CALVIN traverses all of the modes of analysis invoked by the human expert while analyzing ECG data with varying amounts of noise. Many conventional detectors analyze ECGs in a mode that is predominently morphology driven (ie., the beats are classified based on their morphology), regardless of the noise level of the data. The success of this approach in the tests conducted thus far suggests that by modelling human behavior in analyzing noisy ECGs, significant improvement is possible in automated arrhythmia detector performance.

References

[1] B. Lown, A. Calvert, R. Armington, and et. al. Monitoring for serious arrhythmias and high risk of sudden death. *Circulation*, 52:189–198, 1975.

[2] D. C. Harrison, J. W. Fitzgerald, and R. A. Winkle. Ambulatory electrocardiography for diagnosis and treatment of cardiac arrhythmias. *New England Journal of Medicine*, 294:373, 1976.

[3] M. A. Famularo and H. L. Kennedy. Ambulatory electrocardiography in the assessment of pacemaker function. *American Heart Journal*, 104:1086–1094, 1982.

[4] L. A. Vismara, Z. Vera, J. M. Foerster, and et. al. Identification of sudden death risk factors in acute and chronic coronary artery disease. *American Journal of Cardiology*, 821:39, 1977.

[5] E. L. Michelson and J. Morganroth. Spontaneous variability of complex ventricular arrhythmias detected by long-term electrocardiographic recording. *Circulation*, 61:690, 1980.

[6] D. C. Harrison, J. W. Fitzgerald, and R. A. Winkle. Contribution of ambulatory electrocardiographic monitoring to antiarrhythmic management. *American Journal of Cardiology*, 41:996, 1978.

[7] Roger G. Mark and Kenneth L. Ripley. Ambulatory ecg monitoring: real-time analysis versus tape scanning systems. *M.D. Computing*, 2:38–50, 1985.

[8] Roger G. Mark, George B. Moody, W. H. Olson, and et. al. Real-time ambulatory arrhythmia analysis with a microcomputer. In *Computers in Cardiology*, pages 57–62, 1979.

[9] George B. Moody and Roger G. Mark. Development and evaluation of a 2-lead ecg analysis program. In *Computers in Cardiology*, pages 39–44, 1982.

[10] George B. Moody, Warren K. Muldrow, and Roger G. Mark. A noise stress test for arrhythmia detectors. In *Computers in Cardiology*, pages 381–384, 1984.

[11] R. E. Hermes, D. B. Geselowitz, and G. C. Oliver. *Ambulatory Electrocardiographic Recording*, pages 165–181. Yearbook Medical Publishers, Chicago, 1981.

[12] R. E. Hermes, D. B. Geselowitz, and G. C. Oliver. Development, distribution, and use of the american heart association database for ventricular arrhythmia detector evaluation. In *Computers in Cardiology*, pages 263–266, 1980.

[13] George B. Moody and Roger G. Mark. How can we predict real-world performance of an arrhythmia detector. In *Computers in Cardiology*, pages 71–76, 1983.

[14] W. Alraun, C. Zywietz, D. Borovsky, and J. L. Willems. Methods for noise testing of ecg analysis programs. In *Computers in Cardiology*, pages 253–256, 1983.

[15] *MIT-BIH Arrhythmia Database: Tape Directory and Format Specifiation.* Harvard University — Massachusetts Institute of Technology Division of Health Sciences and Technology, Biomedical Engineering Center for Clinical Instrumentation, October 1982. BMEC TR010.

[16] George B. Moody. *EKG Database Applications Programs.* January 1984.

[17] George B. Moody. *Programming Guide to the EKG Databases.* August 1984.

[18] Daniel Chernoff, Tse Lee, George B. Moody, and Roger G. Mark. Evaluation of r-r interval predictors using an annotated ecg database. In *Computers in Cardiology*, pages 359–362, 1981.

[19] Elizabeth M. Allen. *YAPS: Yet Another Production System.* Technical Report TR-1146, University of Maryland, February 1982.

[20] Patrick Henry Winston. *Artificial Intelligence*. Addison-Wesley Publishing Company, Inc., second edition, 1984.

[21] Frederick Hayes-Roth, Donald A. Waterman, and Douglas B. Lenat, editors. *Building Expert Systems*. Addison-Wesley Publishing Company, Inc., 1983.

[22] Lee Brownston, Robert Farrell, Elaine Kant, and Nancy Martin. *Programming Expert Systems in OPS5: An Introduction to Rule-Based Programming*. Addison-Wesley Publishing Company, Inc., 1985.